Key and Inventory Assistant
(KIA)
Software Design Document

Timothy Myers

Monday, March 14th, 2011

Table of Contents

1. Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

2. Decomposition Description 4

2.1 Module and Procedure Decomposition 4

2.2 Data Decomposition 9

3. Dependency Description 13

3.1 Module and Procedure Dependencies 13

3.2 Data Dependencies 17

4. Interface Description 21

4.1 Module and Procedure Interface 21

4.2 Data Interface 38

5. Detailed Design 39

5.1 Module and Procedure Detailed Design 39

5.2 Data Detailed Design 58

1. Introduction

1.1 Purpose

This document has been created to present a design of the modules, processes and data for the KIA project. It will be molding the requirements into a description of the components, modules, and data comprising KIA.

1.2 Scope

The KIA system will work as an administrative aid to Computer Science Department at Tennessee Technological University. It will provide a way for keeping track of the inventory of physical keys in the possession of the department and for managing the groups that have access to certain areas of the department through door swipes. Other than the two functions already listed, the system will also have the ability to track additional simple inventories as those inventories are needed. Currently, the management of all of these items are separated, and the goal of this software is to coalesce the management of these functions into one setting. This should save time and energy for the person managing these items

1.3 Definitions, Acronyms, and Abbreviations

TTU – Tennessee Technological University

CSC Department – Computer Science Department

Groups – A group of people that have access to a set of rooms with their Eagle Card and a card swipe.

1.4 References

MySQL website – www.mysql.com

Qt website – qt.nokia.com

2. Decomposition Description

2.1 Module and Procedure Decomposition

2.1.1 ID: Login Screen

- Type: Module
- Purpose: To allow the user to authenticate with the system and obtain access to the functionality within.
- Function: It will read a username and a password from the user and will check the username and the hash of the given password against what has been stored in the user table. If there is a match, then the user is granted access to the rest of the system.
- Subordinates: none

2.1.2 ID: Database Configuration Screen

- Type: Module
- Purpose: To allow the user to alter the database connection parameters to be used when connecting to the database.
- Function: It will read a hostname, MySQL username, and password from the user and save them to a configuration file. These values will be used in all future attempts to make a connection to the database once the system is restarted.
- Subordinates: none

2.1.3 ID: User Management Screen

- Type: Module
- Purpose: To allow the user to add new users as well as delete and reset the password of existing users.
- Function: In order to add a user, it will read a username and a password from the user and will hash the given password and will attempt to add the user to the users table. In order to reset a password, it will also take a username and a password, and after hashing the password it will attempt to update the password for that user. To delete it simple attempts to delete the record of the selected user from the user table.
- Subordinates: none

2.1.4 ID: People Screen

- Type: Module
- Purpose: To allow the user to add and remove people, search for people, remove a person from one or more groups, select a person to add to a group, select a person to check-out a piece of

inventory, check-in a piece of inventory, and view a set of people in the system.

- Function: It will take input in the form of keyboard entry and mouse actions and will control the people who are stored in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Subordinates: People Addition/Deletion, People Query, Removal of Person from a Group, Inventory Check-in, Add Person to Group, Key/Item Checkout

2.1.5 ID: Room Screen

- Type: Module
- Purpose: To allow the user to add and remove rooms, and view a set of rooms in the system.
- Function: In order to add a room it will take a room name and attempt to add it to the rooms table, in order to remove a room it will attempt to remove that room's record from the rooms table, and it will allow for the information stored on rooms to be viewed.
- Subordinates: None

2.1.6 ID: Keys Screen

- Type: Module
- Purpose: To allow the user to add and remove keys, to search for keys, to begin the process of checking out a key, and to view a set of keys in the system.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the physical keys that are present in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Subordinates: Key Addition/Deletion, Key Query, Key/Item Check-Out

2.1.7 ID: Other Inventory Screen

- Type: Module
- Purpose: To allow the user to add and remove inventory categories, as well as adding and deleting individual items from each category. The user will also be able to search for these items and will be able to initiate an Item check-out from here. It will also allow these items to be viewed by the user.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the inventory categories, and their items that are present in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Subordinates: Category Creation/Deletion, Item Addition/Deletion, Item Query, Key/Item Check-out

2.1.8 ID: Groups Screen

- Type: Module
- Purpose: To allow the user to add and remove groups, remove all people from the group, to begin adding a person to a group, and to view the groups in the system.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the card-key groups that are present in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Subordinates: Group Creation/Deletion, Remove All People from Group, Add Person to Group

2.1.9 ID: Calendar Screen

- Type: Module
- Purpose: To allow the user to view upcoming due dates in the system, and to assign a due date for a check-out.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the due dates that are present in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Subordinates: Due Date Viewing, Key/Item Checkout

2.1.10 ID: People Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove people.
- Function: In order to add a person it will take a T-Number, first name, last name, home street address, home phone number, work phone number, and e-mail address and attempt to add it to the people table, and in order to remove a person it will attempt to remove that person's record from the people table.
- Subordinates: None

2.1.11 ID: People Query

- Type: Procedure
- Purpose: To allow the user to search for people in the system.
- Function: It will be able to take any combination of the input given for user addition and search the people table for people who match the input.
- Subordinates: None

2.1.12 ID: Removal of Person from a Group

- Type: Procedure
- Purpose: To allow the user to remove a person from a group he is a member of.
- Function: It will accept a T-Number and a group ID and will attempt to remove the record linking them in the P-G table.
- Subordinates: None

2.1.13 ID: Inventory Check-in

- Type: Procedure
- Purpose: To allow the user to check a key or other item of inventory back into the system.
- Function: It will accept a T-Number, a Tag Number in the case of a key, or a designation string, in the case of another item, the table name for the type of item, and the amount of that item to return, and will attempt to both decrement the amount checked-out in the record representing the checkout in either the P-K table or the appropriate P-$category table, and update the InInv field of either the Keys table or the appropriate $category table. The check-out record will be removed if the amount checked-out is 0.
- Subordinates: None

2.1.14 ID: Key Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove keys.
- Function: In order to add a key it will take a Tag Number, name, set of rooms, and inventory quantity and attempt to add it to the Keys table or add the quantity amount to the amount in the inventory, and in order to remove a key it will attempt to remove an amount of that key from the available inventory.
- Subordinates: None

2.1.15 ID: Key Query

- Type: Procedure
- Purpose: To allow the user to search for Keys in the system.
- Function: It will be able to take a Tag Number or name and search the Keys table for keys who match the input.
- Subordinates: None

2.1.16 ID: Category Creation/Deletion

- Type: Procedure

- Purpose: To allow the user to add and remove inventory categories.
- Function: In order to add an inventory category it will take a string, CategoryName, and attempt to create a CategoryName table and a P-CategoryName table, and in order to remove a category it will attempt to remove the correct CategoryName and P-CategoryName tables.
- Subordinates: None

2.1.17 ID: Item Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove items to/from a category.
- Function: In order to add an item it will take a descriptor string, a description, a category name, and inventory quantity and attempt to add it to the appropriate $category table or add the quantity amount to the amount in the inventory, and in order to remove an item it will attempt to remove an amount of that item from the available inventory.
- Subordinates: None

2.1.18 ID: Item Query

- Type: Procedure
- Purpose: To allow the user to search for items in the system.
- Function: It will be able to take a descriptor and category name to search the appropriate $category table for items who match the input.
- Subordinates: None

2.1.19 ID: Group Creation/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove key-card groups from the system.
- Function: In order to add a group it will take a group name and a set of rooms and attempt to add it to the Groups table, and in order to remove a group it will attempt to remove that group's record from the Group table.
- Subordinates: None

2.1.20 ID: Remove All People from Group

- Type: Procedure
- Purpose: To allow the user to remove all people from a group.
- Function: It will take a group ID and will attempt to remove all records linking people to that group from the P-G table.

- Subordinates: None

2.1.21 ID: Add Person to Group

- Type: Procedure
- Purpose: To allow the user to add a person to a group.
- Function: It will take a group ID and T-Number and will attempt to add a record linking the referenced person to the group in the P-G table.
- Subordinates: None

2.1.22 ID: Due Date Viewing

- Type: Procedure
- Purpose: To allow the user to view the items due back on a certain date.
- Function: It will take a date and will attempt to query the P-K and all P-$category tables in order to determine any items that are due back on that date. These results are them displayed to the user.
- Subordinates: None

2.1.23 ID: Key/Item Check-out

- Type: Procedure
- Purpose: To allow the user to check a key or other item of inventory out of the system.
- Function: It will accept a T-Number, a Tag Number in the case of a key, or a designation string, in the case of another item, the number of that item to check out and an optional due date, and will attempt to both add the record representing the checkout in either the P-K table or the appropriate P-$category table, and update the InInv field of either the Keys table or the appropriate $category table.
- Subordinates: None

2.2 Data Decomposition

2.2.1 ID: Users Table

- Type: Database Table
- Purpose: To store information concerning the authorized users for KIA. To be accessed for user creation and authentication.
- Function: Contains user information. This includes the username and the hashed password.
- Subordinates: None

2.2.2 ID: People Table

- Type: Database Table
- Purpose: To store information concerning the people in KIA, whether they have checked out inventory, or are part of a Group.
- Function: Contains information on people. This includes: first name, last name, Tnumber, street address, home phone number, work phone number, and e-mail address.
- Subordinates: None

2.2.3 ID: Key Table

- Type: Database Table
- Purpose: To store information concerning the physical keys in the inventory.
- Function: Contains information on keys. This includes: tag number, name, the total number of that key held by the department, and the total number of that key that in currently available for checkout.
- Subordinates: None

2.2.4 ID: P-K Table

- Type: Database Table
- Purpose: To store information concerning which keys have been checked out by which people.
- Function: Contains information on key checkouts. This includes: the tag number of the key, the Tnumber of the person, the number of that key checked out, date the key was checked out, and date the key is due back (if requested).
- Subordinates: None

2.2.5 ID: Groups Table

- Type: Database Table
- Purpose: To store information concerning the Keycard groups for KIA.
- Function: Contains group information. This includes the group's name and a unique group id number.
- Subordinates: None

2.2.6 ID: P-G Table

- Type: Database Table
- Purpose: To store information concerning which people have been assigned to which groups.
- Function: Contains information on group assignments. This includes: the groups id, and the Tnumber of the person.
- Subordinates: None

2.2.7 ID: Rooms Table

- Type: Database Table
- Purpose: To store information concerning Rooms for KIA.
- Function: Contains room information. This includes the room's name and a unique room id number.
- Subordinates: None

2.2.8 ID: K-R Table

- Type: Database Table
- Purpose: To store information concerning which keys open which rooms.
- Function: Contains key ability information. This includes the tag number of the key and the room id number.
- Subordinates: None

2.2.9 ID: G-R Table

- Type: Database Table
- Purpose: To store information concerning which groups are able to access which rooms.
- Function: Contains group ability information. This includes the group id number and the room id number.
- Subordinates: None

Note: The following two tables will be created for each new category of inventory that will be created. I will use $category as a stand in for the category name.

2.2.10 $category Table

- Type: Database Table

- Purpose: To store information concerning the items in $category held in the inventory.
- Function: Contains information on items in $category. This includes: unique designator, description of the object, the total number of that item held by the department, and the total number of that item that in currently available for check out.
- Subordinates: None

2.2.11 P-$category

- Type: Database Table
- Purpose: To store information concerning which items in $category have been checked out by which people.
- Function: Contains information on item checkouts. This includes: the designator for the item, the Tnumber of the person, the number of that item checked out, date the item was checked out, and date the item is due back (if requested).
- Subordinates: None

3. Dependency Description

3.1 Module and Procedure Dependencies

3.1.1 ID: Login Screen

- Type: Module
- Purpose: To allow the user to authenticate with the system and obtain access to the functionality within.
- Dependencies: Read access to the users table.
- Resources : MySQL connection, hashing functionality.

3.1.2 ID: Database Configuration Screen

- Type: Module
- Purpose: To allow the user to alter the database connection parameters to be used when connecting to the database.
- Dependencies: None
- Resources: The system's configuration file.

3.1.3 ID: User Management Screen

- Type: Module
- Purpose: To allow the user to add new users as well as delete and reset the password of existing users.
- Dependencies: User login, Read and write access to the users table.
- Resources: MySQL connection, hashing functionality.

3.1.4 ID: People Screen

- Type: Module
- Purpose: To allow the user to add and remove people, search for people, remove a person from one or more groups, select a person to add to a group, select a person to check-out a piece of inventory, and check-in a piece of inventory.
- Dependencies: User login, People Addition/Deletion, People Query, Removal of Person from a Group, Inventory Check-in, Add Person to Group, Key/Item Checkout.
- Resources: MySQL connection.

3.1.5 ID: Room Screen

- Type: Module
- Purpose: To allow the user to add and remove rooms.
- Dependencies: User login, Read and write access to the rooms table.
- Resources: MySQL connection.

3.1.6 ID: Keys Screen

- Type: Module
- Purpose: To allow the user to add and remove keys, to search for keys, and to begin the process of checking out a key.
- Dependencies: User login, Key Addition/Deletion, Key Query, Key/Item Check-Out.
- Resources: MySQL connection.

3.1.7 ID: Other Inventory Screen

- Type: Module
- Purpose: To allow the user to add and remove inventory categories, as well as adding and deleting individual items from each category. The user will also be able to search for these items and will be able to initiate an Item check-out from here.
- Dependencies: User login, Category Creation/Deletion, Item Addition/Deletion, Item Query, Key/Item Check-out.
- Resources: MySQL connection.

3.1.8 ID: Groups Screen

- Type: Module
- Purpose: To allow the user to add and remove groups, remove all people from the group, and to begin adding a person to a group.
- Dependencies: User login, Group Creation/Deletion, Remove All People from Group, Add Person to Group.
- Resources: MySQL connection.

3.1.9 ID: Calendar Screen

- Type: Module
- Purpose: To allow the user to view upcoming due dates in the system, and to assign a due date for a check-out.

- Dependencies: User login, Due Date Viewing, Key/Item Checkout.
- Resources: MySQL connection.

3.1.10 ID: People Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove people.
- Dependencies: User login, Read and write access to the people table.
- Resources: MySQL connection.

3.1.11 ID: People Query

- Type: Procedure
- Purpose: To allow the user to search for people in the system.
- Dependencies: User login, Read access to the people, P-K, P-G and all P-$category table.
- Resources: MySQL connection.

3.1.12 ID: Removal of Person from a Group

- Type: Procedure
- Purpose: To allow the user to remove a person from a group he is a member of.
- Dependencies: User login, Read and write access to the P-G table.
- Resources: MySQL connection.

3.1.13 ID: Inventory Check-in

- Type: Procedure
- Purpose: To allow the user to check a key or other item of inventory back into the system.
- Dependencies: User login, Read and write access to the Keys and all $category tables, as well as the P-K and all P-$category tables.
- Resources: MySQL connection.

3.1.14 ID: Key Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove keys.
- Dependencies: User login, Read and write access to the Keys table and the Rooms table.
- Resources: MySQL connection.

3.1.15 ID: Key Query

- Type: Procedure
- Purpose: To allow the user to search for Keys in the system.
- Dependencies: User login, Read access to the Keys and K-R tables.
- Resources: MySQL connection.

3.1.16 ID: Category Creation/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove inventory categories.
- Dependencies: User login, Read and write access to the database in order to attempt to create the $category and P-$category tables.
- Resources: MySQL connection.

3.1.17 ID: Item Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove items to/from a category.
- Dependencies: User login, Read and write access to the appropriate $category table.
- Resources: MySQL connection.

3.1.18 ID: Item Query

- Type: Procedure
- Purpose: To allow the user to search for items in the system.
- Dependencies: User login, Read access to the appropriate $category table.
- Resources: MySQL connection.

3.1.19 ID: Group Creation/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove key-card groups from the system.
- Dependencies: User login, Read and write access to the Groups atble.
- Resources: MySQL connection.

3.1.20 ID: Remove All People from Group

- Type: Procedure

- Purpose: To allow the user to remove all people from a group.
- Dependencies: User login, Read and write access to the P-G table.
- Resources: MySQL connection.

3.1.21 ID: Add Person to Group

- Type: Procedure
- Purpose: To allow the user to add a person to a group.
- Dependencies: User login, Read and write access to the P-G table.
- Resources: MySQL connection.

3.1.22 ID: Due Date Viewing

- Type: Procedure
- Purpose: To allow the user to view the items due back on a certain date.
- Dependencies: User login, Read access to the P-K and all P-$category tables.
- Resources: MySQL connection.

3.1.23 ID: Key/Item Check-out

- Type: Procedure
- Purpose: To allow the user to check a key or other item of inventory out of the system.
- Dependencies: User login, Read access to the Keys and all P-$category tables and read and write access to the P-K and all P-$category tables.
- Resources: MySQL connection, physical printer.
-

3.2 Data Dependencies

3.2.1 ID: Users Table

- Type: Database Table
- Purpose: To store information concerning the authorized users for KIA. To be accessed for user creation and authentication.
- Dependencies: Accessed by Login Screen and User management Screen.
- Resources: MySQL Connection

3.2.2 ID: People Table

- Type: Database Table
- Purpose: To store information concerning the people in KIA, whether they have checked out inventory, or are part of a Group.
- Dependencies: Accessed by People Screen, People Addition/Deletion, People Query, and Key/Item Check-out. Referenced by P-K, P-G, and all P-$category tables.
- Resources: MySQL Connection

3.2.3 ID: Key Table

- Type: Database Table
- Purpose: To store information concerning the physical keys in the inventory.
- Dependencies: Accessed by Keys Screen, Inventory Check-in, Key Addition/Deletion, Key Query, Key/Item Check-out. Referenced by P-K, and K-R tables.
- Resources: MySQL Connection

3.2.4 ID: P-K Table

- Type: Database Table
- Purpose: To store information concerning which keys have been checked out by which people.
- Dependencies: Accessed by Inventory Check-In, People Query, Due Date Viewing, and Key/Item Check-Out. References the Key and People tables.
- Resources: MySQL Connection

3.2.5 ID: Groups Table

- Type: Database Table
- Purpose: To store information concerning the Keycard groups for KIA.
- Dependencies: Accessed by Group Creation/Deletion. Referenced by P-G, and G-R tables.
- Resources: MySQL Connection

3.2.6 ID: P-G Table

- Type: Database Table
- Purpose: To store information concerning which people have been assigned to which groups.
- Dependencies: Accessed by People Query, Removal of Person from Group, Remove All People from Group, and Add Person to Group. References the People and Group tables.

- Resources: MySQL Connection

3.2.7 ID: Rooms Table

- Type: Database Table
- Purpose: To store information concerning Rooms for KIA.
- Dependencies: Accessed by Room Screen. Referenced by K-R and G-R tables.
- Resources: MySQL Connection

3.2.8 ID: K-R Table

- Type: Database Table
- Purpose: To store information concerning which keys open which rooms.
- Dependencies: Accessed by Key Query. References the Key and Rooms tables.
- Resources: MySQL Connection

3.2.9 ID: G-R Table

- Type: Database Table
- Purpose: To store information concerning which groups are able to access which rooms.
- Dependencies: Accessed by Groups Screen. References the Groups and Rooms tables.
- Resources: MySQL Connection

Note: The following two tables will be created for each new category of inventory that will be created. I will use $category as a stand in for the category name.

3.2.10 $category Table

- Type: Database Table
- Purpose: To store information concerning the items in $category held in the inventory.
- Dependencies: Modified by Category Creation/Deletion. Accessed by Inventory Check-In, Item Addition/Deletion, and Item Query. Referenced by the matching P-$category table.
- Resources: MySQL Connection

3.2.11 P-$category

- Type: Database Table
- Purpose: To store information concerning which items in $category have been checked out by which people.
- Dependencies: Modified by Category Creation/Deletion. Accessed by People Query, Inventory Check-in, Due Date Viewing, and Key/Item Check-out. References the matching $category table.
- Resources: MySQL Connection

4. Interface Description

4.1 Module and Procedure Interface

4.1.1 ID: Login Screen

- Type: Module
- Purpose: To allow the user to authenticate with the system and obtain access to the functionality within.
- Function: It will read a username and a password from the user and will check the username and the hash of the given password against what has been stored in the user table. If there is a match, then the user is granted access to the rest of the system.
- Interfaces:
  - Inputs:
    - username → alphanumeric string 3 to 20 characters long
    - password → alphanumeric string 5 to 20 characters long
  - Outputs on success:
    - The other functionality of the system is made available to the user.
  - Errors
    - Username is too short.
    - Password is too short.
    - Invalid username and password combination.
    - Database connection and interaction errors.

4.1.2 ID: Database Configuration Screen

- Type: Module
- Purpose: To allow the user to alter the database connection parameters to be used when connecting to the database.
- Function: It will read a hostname, MySQL username, and password from the user and save them to a configuration file. These values will be used in all future attempts to make a connection to the database once the system is restarted.
- Interfaces:
  - Inputs:
    - hostname → a non-empty alphabetic string indicating the hostname of the MySQL server. The default is localhost

    - MySQL username → a non-empty alphanumeric string indicating the username required to access the MySQL server at hostname
    - password → an alphanumeric string indicating the password required by MySQL username to access the MySQL server at hostname. This could be empty if there is no password for the user.
  - Outputs on success:
    - The records for these entries will be updated in the configuration file KIA.conf
  - Errors
    - The configuration file could not be written to.
    - The hostname is empty.
    - The username is empty.

4.1.3 ID: User Management Screen

- Type: Module
- Purpose: To allow the user to add new users as well as delete and reset the password of existing users.
- Function: In order to add a user, it will read a username and a password from the user and will hash the given password and will attempt to add the user to the users table. In order to reset a password, it will also take a username and a password, and after hashing the password it will attempt to update the password for that user. To delete it simple attempts to delete the record of the selected user from the user table.
- Interfaces:
  - Adding a user
    - Inputs:
      - username → an alphanumeric string of 3 to 20 characters
      - password → an alphanumeric string of 5 to 20 characters
    - Outputs on success
      - The password is hashed into a possibly binary string
      - The username and hashed password are inserted into the users table to for a user record.
      - A message is displayed indicating the user was added successfully.
    - Errors
      - The username is too short.

    - The password is too short.
    - Database connection and interaction errors.

- Resetting a Password
  - Inputs:
    - username → an alphanumeric string of 3 to 20 characters
    - password → an alphanumeric string of 5 to 20 characters
  - Outputs on success
    - The password is hashed into a possibly binary string
    - The hashed password is updated in the record for that username in the users table
    - A message is displayed indicating the password was reset successfully.
  - Errors
    - The username is too short.
    - The password is too short.
    - Database connection and interaction errors.
- Deleting a User
  - Inputs:
    - username → an alphanumeric string of 3 to 20 characters
  - Outputs on success
    - The record matching the username is removed from the users table.
    - A message is displayed indicating the user was removed successfully.
  - Errors
    - The username can not be found in the database.
    - Database connection and interaction errors.

4.1.4 ID: People Screen

- Type: Module
- Purpose: To allow the user to add and remove people, search for people, remove a person from one or more groups, select a person to add to a group, select a person to check-out a piece of inventory, check-in a piece of inventory, and view a set of people in the system.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the people who are stored in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the

user.

- Interfaces:
  - Inputs:
    - Keyboard and mouse input will be routed to the appropriate subordinate, which will handle the input.
  - Output on success
    - Successful output of the subordinates will be reflected here.
  - Errors
    - Any errors from a subordinate will be reflected here.

4.1.5 ID: Room Screen

- Type: Module
- Purpose: To allow the user to add and remove rooms, and view a set of rooms in the system.
- Function: In order to add a room it will take a room name and attempt to add it to the rooms table, in order to remove a room it will attempt to remove that room's record from the rooms table, and it will allow for the information stored on rooms to be viewed.
- Interfaces:
  - Adding a Room
    - Inputs
      - Room name → a non-empty alphanumeric string of less than 30 characters
    - Outputs on success
      - A record with the room name is inserted into the rooms table.
      - A message is displayed indicating the room was added successfully.
    - Errors
      - A room exists with that room name in the database.
      - Database connection and interaction errors.
  - Deleting a Room
    - Inputs:
      - Room name → a non-empty alphanumeric string of less than 30 characters
    - Outputs on success
      - The record with the room name is removed from the rooms table.
      - A message is displayed indicating the room was removed successfully.

    - Errors
        - The room name can not be found in the database.
        - Database connection and interaction errors.

4.1.6 ID: Keys Screen

- Type: Module
- Purpose: To allow the user to add and remove keys, to search for keys, to begin the process of checking out a key, and to view a set of keys in the system.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the physical keys that are present in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Interfaces:
    - Inputs:
        - Keyboard and mouse input will be routed to the appropriate subordinate, which will handle the input.
    - Output on success
        - Successful output of the subordinates will be reflected here.
    - Errors
        - Any errors from a subordinate will be reflected here.

4.1.7 ID: Other Inventory Screen

- Type: Module
- Purpose: To allow the user to add and remove inventory categories, as well as adding and deleting individual items from each category. The user will also be able to search for these items and will be able to initiate an Item check-out from here. It will also allow these items to be viewed by the user.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the inventory categories, and their items that are present in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Interfaces:
    - Inputs:
        - Keyboard and mouse input will be routed to the appropriate subordinate, which will handle the input.

    - Output on success
        - Successful output of the subordinates will be reflected here.
    - Errors
        - Any errors from a subordinate will be reflected here.

4.1.8 ID: Groups Screen

- Type: Module
- Purpose: To allow the user to add and remove groups, remove all people from the group, to begin adding a person to a group, and to view the groups in the system.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the card-key groups that are present in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Interfaces:
    - Inputs:
        - Keyboard and mouse input will be routed to the appropriate subordinate, which will handle the input.
    - Output on success
        - Successful output of the subordinates will be reflected here.
    - Errors
        - Any errors from a subordinate will be reflected here.

4.1.9 ID: Calendar Screen

- Type: Module
- Purpose: To allow the user to view upcoming due dates in the system, and to assign a due date for a check-out.
- Function: It will take input in the form of keyboard entry and mouse actions and will control the due dates that are present in the system by delegating the processing of the input to it's subordinates. It will display any output received from the subordinates and display them to the user.
- Interfaces:
    - Inputs:

    - Keyboard and mouse input will be routed to the appropriate subordinate, which will handle the input.
  - Output on success
    - Successful output of the subordinates will be reflected here.
  - Errors
    - Any errors from a subordinate will be reflected here.

4.1.10 ID: People Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove people.
- Function: In order to add a person it will take a T-Number, first name, last name, home street address, home phone number, work phone number, and e-mail address and attempt to add it to the people table, and in order to remove a person it will attempt to remove that person's record from the people table.
- Interfaces:
  - Adding a Person
    - Inputs
      - T-Number → a non-empty alphanumeric string of the form T########
      - first name → a non-empty alphabetic string of less than 21 characters
      - last name → a non-empty alphabetic string of less than 31 characters
      - home address(optional) → a non-empty alphanumeric string of less than 151 characters
      - home phone(optional) → a non-empty string of the form ###-###-####
      - work phone(optional) → a non-empty string of the form ###-###-####
      - e-mail address(optional) → a non-empty string of the form <string>@<string>
    - Outputs on success
      - A record with the given inputs is inserted into the people table.
      - A message is displayed indicating the person was added successfully.
    - Errors
      - The T-Number is not in the correct form.
      - The first name is empty.
      - The last name is empty.
      - (If home phone given) The home phone is not in the correct form.

        - (If work phone given) The work phone is not in the correct form.
        - (If e-mail address given) The e-mail address is not in the correct form.
        - There already exists a person with that T-Number.
        - Database connection and interaction errors.

- Deleting a Person
    - Inputs:
        - T-Number → a non-empty alphanumeric string of the form T########
    - Outputs on success
        - The record with that T-Number is removed from the people table.
        - A message is displayed indicating the person was removed successfully.
    - Errors
        - The T-Number can not be found in the database.
        - Database connection and interaction errors.

4.1.11 ID: People Query

- Type: Procedure
- Purpose: To allow the user to search for people in the system.
- Function: It will be able to take any combination of the input given for user addition and search the people table for people who match the input.
- Interfaces:
    - Inputs
        - T-Number(optional) → a non-empty alphanumeric string
        - first name(optional) → a non-empty alphabetic string of less than 21 characters
        - last name(optional) → a non-empty alphabetic string of less than 31 characters
        - home address(optional) → a non-empty alphanumeric string of less than 151 characters
        - home phone(optional) → a non-empty string
        - work phone(optional) → a non-empty string
        - e-mail address(optional) → a non-empty string
    - Outputs on success
        - A list of people who match the given inputs using the MySQL LIKE operator are displayed, or 'No match' if no one matches the input.

        - Errors
            - (If no inputs are given) The query can not be preformed.
            - Database connection and interaction errors.

4.1.12 ID: Removal of Person from a Group

- Type: Procedure
- Purpose: To allow the user to remove a person from a group he is a member of.
- Function: It will accept a T-Number and a group ID and will attempt to remove the record linking them in the P-G table.
- Interface:
    - Inputs
        - T-Number → a non-empty alphanumeric string of the form T########
        - Group ID → a positive integer <= 2147483647
    - Outputs on success
        - The record indicating that the person is in the specified group is removed from the P-G table.
        - A message is displayed indicating the person was successfully removed from the group.
    - Errors
        - That person was not a member of the given group.
        - Database connection and interaction errors.

4.1.13 ID: Inventory Check-in

- Type: Procedure
- Purpose: To allow the user to check a key or other item of inventory back into the system.
- Function: It will accept a T-Number, a Tag Number in the case of a key, or a designation string, in the case of another item, the table name for the type of item, and the amount of that item to return, and will attempt to both decrement the amount checked-out in the record representing the checkout in either the P-K table or the appropriate P-$category table, and update the InInv field of either the Keys table or the appropriate $category table. The check-out record will be removed if the amount checked-out is 0.
- Interfaces:
    - Inputs

        - T-Number → a non-empty alphanumeric string of the form T########
        - Tag Number → positive integer <= 2147483647

          or
        - Designator → a non-empty alphanumeric string of less than 50 characters
        - Table →string, either 'Keys' or a category name indicating which type of item
        - Amount to return → positive integer <= the amount the person has checked-out

    - Outputs on success
        - The record indicating that the person has checked-out the inventory item is modified by subtracting the amount to return from what they have checked-out in the P-K or P-$category table.
        - If the amount they have checked-out is zero the record is removed from the P-K or P-$category table.
        - The amount of that item that is available to be checked-out is increased by the amount to return in the Keys or $category table.
        - A message is displayed indicating the item was successfully checked-in to the inventory.
    - Errors
        - The person does not have any of that item checked-out.
        - The person does not have at least amount to return of that item checked-out.
        - Database connection and interaction errors.

4.1.14 ID: Key Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove keys.
- Function: In order to add a key it will take a Tag Number, name, set of rooms, and inventory quantity and attempt to add it to the Keys table or add the quantity amount to the amount in the inventory, and in order to remove a key it will attempt to remove an amount of that key from the available inventory.
- Interfaces:
    - Adding a Key
        - Inputs
            - Tag Number → positive integer <= 2147483647
            - name → a non-empty alphanumeric string of less than 31 characters
            - set of room names → comma separated list of room names

    - inventory quantity →positive integer <= 2147483647
  - Outputs on success
    - If the Key is not in the system, it is added.
    - If the Tag Number is found in the system, then the inventory quantity is added to the current total inventory.
    - If any of the rooms listed are not in the system, they are added to the Rooms table.
    - A record is added to the Keys table with the given name, Tag Number and initial inventory.
    - Records are inserted into the K-R table to link the newly added key to the list of rooms given.
    - A message is displayed indicating the key was added successfully.
  - Errors
    - The Tag Number is negative.
    - The Tag Number is too large.
    - The Tag Number can not be found in the database.
    - The name is empty.
    - The initial inventory is negative.
    - The initial inventory is too large.
    - There already exists a key with that Tag Number.
    - Database connection and interaction errors.
- Deleting a Key
  - Inputs:
    - Tag Number → positive integer <= 2147483647
    - amount to remove → positive integer <= amount of key available to check-out
  - Outputs on success
    - The record with that Tag Number is updated by subtracting the amount to remove from both the total inventory and the amount available to check-out in the Keys table.
    - If the total inventory for that key reaches zero, that record is removed from the Keys table.
    - A message is displayed indicating the key was removed successfully.
  - Errors
    - The Tag Number is negative.

        - The Tag Number is too large.
        - The Tag Number can not be found in the database.
        - The amount to remove is more than the amount available to check-out.
        - Database connection and interaction errors.

4.1.15 ID: Key Query

- Type: Procedure
- Purpose: To allow the user to search for Keys in the system.
- Function: It will be able to take a Tag Number or name and search the Keys table for keys who match the input.
- Interfaces:
    - Inputs
        - Tag Number(optional) → positive integer <= 2147483647
        - name(optional) → a non-empty alphanumeric string of less than 31 characters
        - set of room names(optional) → comma separated list of room names
    - Outputs on success
        - A list of keys who match the given Tag Number and name using the MySQL LIKE operator, and are connected to all of the given rooms are displayed, or 'No match' if no key matches the input.
    - Errors
        - (If no inputs are given) The query can not be preformed.
        - Database connection and interaction errors.

4.1.16 ID: Category Creation/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove inventory categories.
- Function: In order to add an inventory category it will take a string, CategoryName, and attempt to create a CategoryName table and a P-CategoryName table, and in order to remove a category it will attempt to remove the correct CategoryName and P-CategoryName tables.
- Interface:
    - Creating a Category
        - Inputs
            - CategoryName → a non-empty alphanumeric string of less then 31 characters

    - Outputs on success
        - A table named CategoryName will be created to hold item of that type.
        - A table named P-CategoryName will be created to hold information about who has those items checked-out.
        - A message is displayed indicating the Category was successfully added to the database.
    - Errors
        - Database connection and interaction errors.
- Deleting a Category
    - Inputs
        - CategoryName → a non-empty alphanumeric string of less then 31 characters
    - Outputs on success
        - The table named CategoryName will be deleted from the database.
        - The table named P-CategoryName will be deleted from the database.
        - A message is displayed indicating the Category was successfully deleted from the database.
    - Errors
        - Database connection and interaction errors.

4.1.17 ID: Item Addition/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove items to/from a category.
- Function: In order to add an item it will take a descriptor string, a description, a category name, and inventory quantity and attempt to add it to the appropriate $category table or add the quantity amount to the amount in the inventory, and in order to remove an item it will attempt to remove an amount of that item from the available inventory.
- Interfaces:
    - Adding an Item
        - Inputs
            - descriptor → a non-empty alphanumeric string of less then 51 character
            - description(optional) → a non-empty alphanumeric string of less then 256 character
            - inventory quantity→ positive integer <= 2147483647
            - category name → non-empty string indicating the item's category

- Outputs on success
    - If the descriptor does not exist in the database, a record is added to the appropriate $category table with the given descriptor, description and initial inventory.
    - If the descriptor does exist in the database, the amount in total inventory and available to check out is increased by inventory quantity.
    - A message is displayed indicating the item was added successfully.
- Errors
    - The descriptor is empty.
    - The initial inventory is negative.
    - The initial inventory is too large.
    - There already exists an item with that descriptor.
    - Database connection and interaction errors.

- Deleting an Item
    - Inputs:
        - descriptor → a non-empty alphanumeric string of less then 51 character
        - amount to remove → positive integer <= amount of that item available to check-out
        - category name → non-empty string indicating the item's category
    - Outputs on success
        - The record with that descriptor is updated by subtracting the amount to remove from both the total inventory and the amount available to check-out in the $category table.
        - If the total inventory for that item reaches zero, that record is removed from the $category table.
        - A message is displayed indicating the item was removed successfully.
    - Errors
        - The descriptor is empty
        - The descriptor can not be found in the database.
        - The amount to remove is more than the amount available to check-out.
        - Database connection and interaction errors.

4.1.18 ID: Item Query

- Type: Procedure

- Purpose: To allow the user to search for items in the system.
- Function: It will be able to take a descriptor and category name to search the appropriate $category table for items who match the input.
- Interfaces:
  - Inputs
    - descriptor(optional) → a non-empty alphanumeric string of less then 51 character
    - category name → non-empty string indicating the item's category
  - Outputs on success
    - A list of items who match the given descriptor using the MySQL LIKE operator is displayed, or 'No match' if no item matches the input.
  - Errors
    - (If descriptor is not given) The query can not be preformed.
    - Database connection and interaction errors.

4.1.19 ID: Group Creation/Deletion

- Type: Procedure
- Purpose: To allow the user to add and remove key-card groups from the system.
- Function: In order to add a group it will take a group name and a set of rooms and attempt to add it to the Groups table, and in order to remove a group it will attempt to remove that group's record from the Group table.
  - Creating a Group
    - Inputs
      - Group name → a non-empty alphanumeric string of less than 31 characters
      - set of room names → comma separated list of room names
    - Outputs on success
      - If any of the rooms listed are not in the system, they are added to the Rooms table.
      - A record is added to the groups table with the given group name.
      - Records are inserted into the G-R table to link the newly added group to the list of rooms given.
      - A message is displayed indicating the key was added successfully.
    - Errors
      - The group name is empty.
      - There already exists a group with that group name.

        - Database connection and interaction errors.
    - Deleting a Group
        - Inputs:
            - Group name → a non-empty alphanumeric string of less than 31 characters
        - Outputs on success
            - The record containing that group name is removed from the groups table.
            - A message is displayed indicating the group was removed successfully.
        - Errors
            - The group name can not be found in the system.
            - Database connection and interaction errors.

4.1.20 ID: Remove All People from Group

- Type: Procedure
- Purpose: To allow the user to remove all people from a group.
- Function: It will take a group ID and will attempt to remove all records linking people to that group from the P-G table.
- Interface:
    - Inputs
            - Group ID → a positive integer <= 2147483647
    - Outputs on success
        - All records linking this group to people are removed from the P-G table.
        - A message is displayed indicating the people were successfully removed from the group.
    - Errors
        - That Group ID does not exist in the system
        - Database connection and interaction errors.

4.1.21 ID: Add Person to Group

- Type: Procedure
- Purpose: To allow the user to add a person to a group.
- Function: It will take a group ID and T-Number and will attempt to add a record linking the referenced person to the group in the P-G table.

- Interface:
  - Inputs
    - T-Number → a non-empty alphanumeric string of the form T########
    - Group ID → a positive integer <= 2147483647
  - Outputs on success
    - The record indicating that the person is in the specified group is added to the P-G table.
    - A message is displayed indicating the person was successfully added to the group.
  - Errors
    - That T-Number was not found in the system.
    - That Group ID was not found in the system.
    - Database connection and interaction errors.

4.1.22 ID: Due Date Viewing

- Type: Procedure
- Purpose: To allow the user to view the items due back on a certain date.
- Function: It will take a date and will attempt to query the P-K and all P-$category tables in order to determine any items that are due back on that date. These results are them displayed to the user.
- Interfaces:
  - Inputs
    - date → a date in the form MM/DD/YYYY
  - Outputs on success
    - A list of items and people whose items are due back on the selected date is displayed, or 'No item due back' if no date matches the input.
  - Errors
    - Invalid date was given.
    - Database connection and interaction errors.

4.1.23 ID: Key/Item Check-out

- Type: Procedure
- Purpose: To allow the user to check a key or other item of inventory out of the system.
- Function: It will accept a T-Number, a Tag Number in the case of a key, or a designation string, in the case of another item, the number of that item to check out and an optional due date, and

will attempt to both add the record representing the checkout in either the P-K table or the appropriate P-$category table, and update the InInv field of either the Keys table or the appropriate $category table.

- Interfaces:
  - Inputs
    - T-Number → a non-empty alphanumeric string of the form T########
    - Tag Number → positive integer <= 2147483647

      or
    - Designator → a non-empty alphanumeric string of less than 50 characters
    - Amount to check-out → positive integer <= the amount available to check-out
    - due date(optional) → a date in the form MM/DD/YYYY
  - Outputs on success
    - The record containing the T-Number, Tag Number/Descriptor, amount checked-out and the due date(if given) indicating that the person has checked-out the inventory item is added to the P-K or P-$category table.
    - The amount of that item that is available to be checked-out is decreased by the amount to check-out.
    - A message is displayed indicating the item was successfully checked-out of the inventory.
  - Errors
    - The person does not exist in the database.
    - The Key does not exist in the database

      or
    - The item does not exist in the database.
    - The date is invalid.
    - The amount to check-out is more than what is available to check-out.
    - The amount to check-out is negative.
    - Database connection and interaction errors.

4.2 Data Interface

All data stored in the database tables will be accessed through the SQL interface provided my MySQL.

5. Detailed Design

5.1 Module and Procedure Detailed Design

5.1.1 ID: Login Screen

- Processing:

  If the username does not meet the validation requirements, an error message is displayed, or of the password does not meet the validation requirement, an error message is displayed. If both are validated then the password is hashed and both the username and password are used to query the users table in the database. If an error comes from the database connection then an appropriate error message is displayed. If there is one record containing the username and password, then the user has been verified and the user is shown the People Screen and given access to all other system functionality. If there is not one record containing the username and password, then the user has not been verified so the user remains at the Login Screen and an error message is displayed to the user.

- Data
  - Hash

    The given password is split in half and then the given username is concatenated into the middle of the password and hashed. This hash result is then hashed again using a different hashing algorithm. The result of this second hash is the hash used to check against the database.

  - Error String

    The string that is displayed to the user if the username or password do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.2 ID: Database Configuration Screen

- Processing:

  If the hostname or the MySQL username are empty then an error message is displayed to the user and all further processing stops. The system attempts to open the file KIA.conf for reading and the the attempt to open the file KIA.conf.tmp for writing is made. If either of these openings fail then an error message is displayed to the user and further processing is stopped. The lines in KIA.conf are read and if they do not begin with the 'hname', 'MySQLu', or 'MySQLp' tags then they are written out exactly to KIA.conf.tmp as they are read. If they do match one of those tags however, then those tags are written out to KIA.conf.tmp as 'tagName:varName'. Once the new file has been written, KIA.conf will be deleted, and KIA.conf.tmp will be renamed to KIA.conf.

- Data
  - OldConf

    A file pointer that points to KIA.conf. It holds the old database configuration for the system. It will be read to repopulate NewConf.
  - NewConf

    A file pointer that points to KIA.conf.tmp. It is populated with the new database configuration information and all other information read from OldConf.
  - Success String

    The string that is displayed to the user if the configuration was successfully changed. It's default value is 'Your new database configuration has been saved.' The text will be black when displayed to the user.
  - Error String

    The string that is displayed to the user if the username or password do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.3 ID: User Management Screen

- Adding a user
  - Processing:

    If the username or the password is too short an error message is displayed to the user and further processing stops. The system will not allow the user to input more than 20 characters for the username or password. If both username and password given match the data requirements then the password is hashed and there is an attempt to add the given username and the hash of the given password to the users table. An error message is displayed if there is already a user with that username, and any other errors that come from the database connection will be reported to the user as well. If the user was added successfully then a success message is displayed to the user.
  - Data
    - Hash

      The given password is split in half and then the given username is concatenated into the middle of the password and hashed. This hash result is then hashed again using a different hashing algorithm. The result of this second hash is the hash used in attempting to add the user.

    - Success String

      The string that is displayed to the user if the user was successfully added. It's default value is 'That user has been added to the system.' The text will be black when displayed to the user.

    - Error String

      The string that is displayed to the user if the username or password do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

- Resetting a Password
  - Processing:

    If the username or the password is too short an error message is displayed to the user and further processing stops. The system will not allow the user to input more than 20 characters for the username or password. If both username and password given match the data requirements then the password is hashed and there is an attempt to update the record of the the given username with the hash of the given password in the users table. An error message is displayed if there is no user with that username, and any other errors that come from the database connection will be reported to the user as well. If the password was reset successfully then a success message is displayed to the user.

  - Data
    - Hash

      The given password is split in half and then the given username is concatenated into the middle of the password and hashed. This hash result is then hashed again using a different hashing algorithm. The result of this second hash is the hash used in the attempt to change the password.

    - Success String

      The string that is displayed to the user if the user was successfully added. It's default value is 'The password for 'USERNAME' has been reset.' The text will be black when displayed to the user.

    - Error String

      The string that is displayed to the user if the username or password do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

- Deleting a User
  - Processing:

    A username is accepted and an attempt is made to remove the record with the given username from the users table. If there is no user with that user name then an error message is displayed. The same will happen if the database connection produces any errors. If the record is deleted then the success string is displayed.

- Data
  - Success String

    The string that is displayed to the user if the user was successfully added. It's default value is 'That user has been removed from the system.' The text will be black when displayed to the user.
  - Error String

    The string that is displayed to the user if the username or password do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.4 ID: People Screen

- Processing:

  This module accepts all of the input required by its subordinates from the user and then hands them to the requested subordinate with minimal processing.

  For People Addition and People Query this module ensures that text input with a maximum length does not exceed that maximum length and also discovers which optional arguments are given.

  For People Deletion, Adding a person to a group, and Key/Item Check-out this module extracts the T-Number from the selected person in the list of people found by People Query.

  For Removing a Person from a Group this module extracts the Group ID from the Person's Group List.

  For Checking-In Inventory this module extracts the Tag Number or Designator from the Person's Inventory List of items the user has checked-out and then accepts the amount to check in.

  For Viewing a person this module will populate the Person's Group List by searching the a joining of the Groups table and the P-G table, it will populate the the Person's Inventory List by searching a join between the P-K and Keys table and all P-$category tables. It will then create the Selected Person string with data from the People table and the Person's Group List and the Person's Inventory List. The Selected Person String will then be displayed for viewing. Any errors that come from the database connection will be displayed to the user and will cause the Selected Person string to be set to its default.

  Once this input has been given to the chosen subordinate, this module waits for the subordinate to produce output which it either displays or stores internally.

- Data
  - People List
    - The list of people returned from the People Query subordinate. This list contains the T-Number, first, and last name of the people that match the query directed at People Query. It is by default empty.
  - Person's Group List
    - The list of groups that the selected person is in. This list includes the Group name and Group ID for each group that the selected user is a member of. It is by default empty.
  - Person's Inventory List
    - The list of items that the selected person has checked-out. This list includes the Tag Number, Key name, and string 'Keys' for keys/ the Descriptor and string '$category' for other inventory and the amount of that item the selected person has checked-out.
  - Selected Person
    - A string which displays information on the selected person. This information includes all of the data in the People table for that person as well as the groups that the person is in and the inventory items he has checked-out. It's default value is 'No person selected'.
  - Subordinate String Output
    - A string which will hold either success or error messages as returned by the subordinates. To be displayed.

5.1.5 ID: Room Screen

- Adding a Room
  - Processing:

    If the is empty an error message is displayed to the user and further processing stops. The system will not allow the user to input more than 30 characters for the room name. If room name match the data requirements then there is an attempt to add the given room name to the rooms table. An error message is displayed if there is already a room with that room name, and any other errors that come from the database connection will be reported to the user as well. If the room was added successfully then a success message is displayed to the user.
  - Data
    - Success String

      The string that is displayed to the user if the room was successfully added. It's default value is 'That room has been added to the system.' The text will be black when displayed to the user.
    - Error String

      The string that is displayed to the user if the room name does not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

- Deleting a Room
  - Processing:

    A room id is accepted and an attempt is made to remove the record with the given room name from the rooms table. If successful this delete should cascade through the K-R and G-R tables. If there is no room with that room id then an error message is displayed. The same will happen if the database connection produces any errors. If the record is deleted then the success string is displayed.
  - Data
    - Success String

      The string that is displayed to the user if the room was successfully removed. It's default value is 'That room has been removed from the system.' The text will be black when displayed to the user.
    - Error String

      The string that is displayed to the user if the room id do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.
- Room Viewing
  - Processing:

    For Viewing a person this module will populate the Selected Room string by retrieving the room name from the selected item in the Rooms List. It will also search a join of the Keys and K-R and a join of the Groups and G-R tables to find the Group Names and Key Names to complete the Selected Room string. Any errors that come from the database connection will be displayed to the user and will cause the Selected Person string to be set to its default.
  - Data
    - Selected Room

      A string which displays information on the selected room. This information includes the room name as well as which keys open this room and which groups have access to this room. It's default value is 'No room selected'.
    - Rooms List

      The list of rooms in the system. This list contains the Room Name and Room ID of all the rooms in the system. It is by default empty.

5.1.6 ID: Keys Screen

- Processing:

This module accepts all of the input required by its subordinates from the user and then hands them to the requested subordinate with minimal processing.

For Key Addition and Key Query this module ensures that text input with a maximum length does not exceed that maximum length and also discovers which optional arguments are given.

For Key Deletion and Key/Item Check-out this module extracts the Tag Number from the selected Key in the Key List.

For Key/Item Check-out this module accepts an amount of the key to check-out from the user, as well as asking the user whether or not the user wishes for a due date. This input is stored globally for later use.

For Viewing a key this module will populate the Selected Key string by extracting the Key name from the Key List as well as searching a join between the K-R and Rooms tables and a join between the P-R and People table to complete the Selected Key string. The Selected Key string will then be displayed for viewing. Any errors that come from the database connection will be displayed to the user and will cause the Selected Key string to be set to its default.

Once this input has been given to the chosen subordinate, this module waits for the subordinate to produce output which it either displays or stores internally.

- Data
  - Key List
    - The list of keys returned from the Key Query subordinate. This list contains the Tag number, and Key name of the keys that match the query directed at Key Query. It is by default empty.
  - Selected Key
    - A string which displays information on the selected key. This information includes the Tag Number and Key name as well as the rooms that the key can open and people who have at least one of this type of key checked-out. It's default value is 'No key selected'.
  - Subordinate String Output
    - A string which will hold either success or error messages as returned by the subordinates. To be displayed.

5.1.7 ID: Other Inventory Screen

- Processing:

This module accepts all of the input required by its subordinates from the user and then hands them to the requested subordinate with minimal processing.

For Category Creation, Item Addition and Item Query this module ensures that text input with a maximum length does not exceed that maximum length and also discovers which optional arguments are given.

For Category Deletion this module extracts the category name from the selected Category in the

Category list.

For Item Deletion and Key/Item Check-out this module extracts the Descriptor from the selected item in the Item List.

For Key/Item Check-out this module accepts an amount of the item to check-out from the user, as well as asking the user whether or not the user wishes for a due date. This input is stored globally for later use.

For Viewing an item this module will populate the Selected Item string by extracting the Descriptor and Description from the $category table as searching a join between the P-$category table and People tables complete the Selected Key string. The Selected Key string will then be displayed for viewing. Any errors that come from the database connection will be displayed to the user and will cause the Selected Key string to be set to its default.

Once this input has been given to the chosen subordinate, this module waits for the subordinate to produce output which it either displays or stores internally.

- Data
  - Category List
    - The list of all categories in the system. This list contains the category names of all categories in the system. It is by default empty.
  - Item List
    - The list of items returned from the Item Query subordinate. This list contains the Descriptors of the items that match the query directed at Item Query. It is by default empty.
  - Selected Item
    - A string which displays information on the selected items. This information includes the item Descriptor and Description as well as the people who have at least one of this type of item checked-out. It's default value is 'No item selected'.
  - Subordinate String Output
    - A string which will hold either success or error messages as returned by the subordinates. To be displayed.

5.1.8 ID: Groups Screen

- Processing:

  This module accepts all of the input required by its subordinates from the user and then hands them to the requested subordinate with minimal processing.

  For Group Creation this module ensures that text input with a maximum length does not exceed that maximum length.

  For Group Deletion, Remove all People from Group, and Add Person to Group this module extracts the Group ID from the selected group in the Group List.

For Viewing a group this module will populate the Selected Group string by extracting the Group name from the Group List as well as searching a join between the K-R and Rooms tables and a join between the P-R and People table to complete the Selected Key string. The Selected Group string will then be displayed for viewing. Any errors that come from the database connection will be displayed to the user and will cause the Selected Key string to be set to its default.

Once this input has been given to the chosen subordinate, this module waits for the subordinate to produce output which it either displays or stores internally.

- Data
  - Group List
    - The list of all groups in the system. This list contains the Group ID, and Group name of tall groups in the system. It is by default empty.
  - Selected Group
    - A string which displays information on the selected group. This information includes the Group name as well as the rooms that the group can access and people who are in the group. It's default value is 'No group selected'.
  - Subordinate String Output
    - A string which will hold either success or error messages as returned by the subordinates. To be displayed.

5.1.9 ID: Calendar Screen

This module accepts all of the input required by its subordinates from the user and then hands them to the requested subordinate with minimal processing.

For Due Date Viewing and Item/Key Check-out this module ensures that date input is a valid date.

For Viewing a Due Date this module will display the list returned by Due Date Viewing to show the people who have an item due back and the Key name or item descriptor of the item due back.

Once this input has been given to the chosen subordinate, this module waits for the subordinate to produce output which it either displays or stores internally.

- Data
  - Due Back List
    - The list returned by Due Date Viewing. This list contains the last name, first name, and T-Number of the person borrowing, Key Name/Descriptor, and category name(could be 'Key')of the item to be returned. It is by default empty.

- Subordinate String Output
    - A string which will hold either success or error messages as returned by the subordinates. To be displayed.

5.1.10 ID: People Addition/Deletion

- Adding a Person
    - Processing:

      An error message will be generated and output and processing stopped for a number of reasons. These include: T-Number not in the correct format, first name empty, last name empty, either phone number not in the correct format(if either are given), and the e-mail address is not in the correct format(if it is given). If given input match the data requirements then there is an attempt to add the given inputs to the people table. An error message is displayed if there is already a person with the given T-Number, and any other errors that come from the database connection will be reported to the user as well. If the person was added successfully then a success message is displayed to the user.
    - Data
        - Success String

          The string that is displayed to the user if the person was successfully added. It's default value is 'The person has been added to the system.' The text will be black when displayed to the user.
        - Error String

          The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

- Removing a Person
    - Processing:

      A T-Number is accepted and an attempt is made to remove the record with the given T-Number from the rooms table. If the T-Number is not in the correct form then an error message is displayed and processing stops. If successful this delete should cascade through the P-K and all P-$category tables. If there is no person with that T-Number then an error message is displayed. The same will happen if the database connection produces any errors. If the record is deleted then the success string is displayed.
    - Data
        - Success String

          The string that is displayed to the user if the person was successfully removed. It's default value is 'That person has been removed from the system.' The text will be black when displayed to the user.

- Error String

  The string that is displayed to the user if the T-Number do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.11 ID: People Query

- Processing:

  If all arguments to People Query are empty then an error is returned and processing stops. Otherwise the people table is queried with all given parameters using the MySQL LIKE operator in such a manner that if the input was anywhere in the record it would be returned. The list of people returned would be added to the People List and returned. Any errors coming from the database connection would cause an error message to be returned.

- Data

  - People List

    The list of people to return. This list will contain the T-Number, Last name, and first name of all people who matched the input given to People query. It is empty by default.

  - Error String

    The string that is displayed to the user if the input is empty or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.12 ID: Removal of Person from a Group

- Processing:

  If the T-Number is not in the correct form or if the Group ID is outside of the acceptable range then an error message is returned and no further processing is done. If the input match the data requirements then there is an attempt to remove the record from the P-G table indicating the person's membership in the group. An error message is displayed if there is no matching record in the database, and any other errors that come from the database connection will be reported to the user as well. If the person was removed from the group successfully then a success message is returned.

- Data

  - Success String

    The string that is displayed to the user if the person was successfully removed from the group. It's default value is 'The person has been removed from that group.' The text will be black when displayed to the user.

  - Error String

    The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.13 ID: Inventory Check-in

- Processing:

  If the T-Number is not in the correct form, the Tag Number is outside of the acceptable range/Descriptor is empty, the person does not have any of that item checked out, the person has less than they want to check-in checked-out, or the table indicating which type of item does not exist, then an error message will be returned and further processing will stop. If the input match the data requirements then there is an attempt to decrement the amount the user has checked out by the amount being checked-in, or to remove the record if the person is checking-in all of that item. This will occur in either the P-K table or one of the P-$category tables. The amount of inventory available to check-out in either the Keys table or one of the $category tables for that item is also incremented by the amount returned. An error message is returned if there is no matching T-Number, Tag Number, or descriptor in the database, and any other errors that come from the database connection will be reported to the user as well. If the inventory was checked-in successfully then a success message is returned.

- Data

  - Success String

    The string that is displayed to the user if the inventory was successfully checked-in. It's default value is 'The check-in was successful.' The text will be black when displayed to the user.

  - Error String

    The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.14 ID: Key Addition/Deletion

- Adding a Key

  - Processing:

    If the Tag Number is outside the valid range, the initial inventory is outside the valid range, or the name is empty then an error message will be generated and returned and processing stopped.

    If the Tag Number is in the database, then the Key name and room list is ignored and an attempt is made to increment the total inventory of that key and total available to check-out by the inventory quantity given.

    If the Tag Number is not in the database there is an attempt to parse the room names. A database transaction is started. The string is split along commas and each item is added to the rooms table, duplicate errors are ignored but any other database errors cause an error string to be generated, the transaction rolled back and the message returned. Once the rooms have been added the other inputs are inserted into the Keys table. The transaction is rolled back and an error message is returned if there is already a Key with the given Tag Number. Then rows are inserted into the K-R table for each room given to associate the key with those rooms. Any errors that come from the database connection will cause the transaction to be rolled back and an error to be returned to the user as well. If the item was added successfully then the transaction is committed and a success message is returned to the user.

- Data
    - Room list

      A list of string which contain the room names to be associated with this key.

    - Success String

      The string that is displayed to the user if the key was successfully added. It's default value is 'Those keys have been added to the system.' The text will be black when displayed to the user.

    - Error String

      The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

- Removing a Key
    - Processing:

      If the Tag-Number is outside the valid range or the amount to remove is more then what is available to check-out, then an error message will be returned and further processing will stop. If the input match the data requirements then there is an attempt to decrease the amount available for check-out and the total inventory amount by the amount in the input. If this decrease would lower the total inventory to zero, then the key's record is deleted from the Keys table instead. These deletion will cascade through the P-K and K-R tables. An error message is returned if there is no matching Tag Number in the database, and any other errors that come from the database connection will be reported to the user as well. If the key was removed successfully then a success message is returned.

    - Data
        - Success String

          The string that is displayed to the user if the key was successfully removed. It's default value is 'Those keys have been removed from the system.' The text will be black when displayed to the user.

        - Error String

          The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.15 ID: Key Query

- Processing:

  If all arguments to Key Query are empty then an error is returned and processing stops. Otherwise the Keys table is queried with the Tag Number, and key name parameters if given using the MySQL LIKE operator in such a manner that if the input was anywhere in the record it would be returned. The string of rooms is parsed and the K-R table is searched to get a list of rooms that can open all given keys. The list of rooms found in the results of both previously mentioned queries will be added to the Key List and returned. Any errors coming from the database connection would cause an error message to be returned.

- Data

  - Key List

    The list of keys to return. This list will contain the Tag Name and Key name of all keys who matched the input given to Key query. It is empty by default.

  - Error String

    The string that is displayed to the user if the input is empty or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.16 ID: Category Creation/Deletion

- Adding a category

  - Processing:

    If the category name is empty an error message is displayed to the user and further processing stops. If the category name matches the data requirements then there is an attempt to add a table of with the name of category name and another table with a name of P-category name. These tables are created to the format specified in the data design. An error message is returned if there are already tables with those names, as this means that the category already exists, and any other errors that come from the database connection will be returned to the user as well. If the category was added successfully then a success message is displayed to the user.

  - Data

    - Success String

      The string that is displayed to the user if the category was successfully added. It's default value is 'The category has been added to the system.' The text will be black when displayed to the user.

    - Error String

      The string that is displayed to the user if the category already exists or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

- Deleting a category
  - Processing:

    A category name is accepted and if it is not empty an attempt is made to remove the tables names category name and P-category name from the database. If there is no category with that name then an error message is returned. The same will happen if the database connection produces any errors. If the category is deleted then the success string is displayed.
  - Data
    - Success String

      The string that is displayed to the user if the category was successfully added. It's default value is 'The category has been removed from the system.' The text will be black when displayed to the user.
    - Error String

      The string that is displayed to the user if the category does not exist or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.17 ID: Item Addition/Deletion

- Adding an Item
  - Processing:

    If the Descriptor is empty, the inventory quantity is outside the valid range, the category name is empty, or a table named category name does not exist in the database then an error message will be generated and returned and processing stopped.

    If the Descriptor is in the table named category name, then the description is ignored and an attempt is made to increment the total inventory of that item and total available to check-out by the inventory quantity given.

    If the Descriptor is not in the database there is an attempt to add the inputs into the category table. Any errors that come from the database connection will cause an error to be returned to the user. If the item was added successfully then a success message is returned to the user.
  - Data
    - Success String

      The string that is displayed to the user if the item was successfully added. It's default value is 'Those items have been added to the system.' The text will be black when displayed to the user.
    - Error String

      The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

- Removing a Key
  - Processing:

    If the Descriptor is empty, the category name is empty, there does not exist a table named category name, or the amount to remove is more then what is available to check-out, then an error message will be returned and further processing will stop. If the input match the data requirements then there is an attempt to decrease the amount available for check-out and the total inventory amount by the amount in the input. If this decrease would lower the total inventory to zero, then the item's record is deleted from the category name table instead. An error message is returned if there is no matching Descriptor in the category name table, and any other errors that come from the database connection will be reported to the user as well. If the item was removed successfully then a success message is returned.
  - Data
    - Success String

      The string that is displayed to the user if the item was successfully removed. It's default value is 'Those items have been removed from the system.' The text will be black when displayed to the user.
    - Error String

      The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.18 ID: Item Query

- Processing:

  If the Descriptor is empty then an error is returned and processing stops. Otherwise the category name table is queried with the Descriptor parameters if given using the MySQL LIKE operator in such a manner that if the input was anywhere in the record it would be returned. These values will be added to the Item List and returned. Any errors coming from the database connection would cause an error message to be returned.
- Data
  - Item List

    The list of keys to return. This list will contain the Descriptors of all items who matched the input given to Item Query. It is empty by default.
  - Error String

    The string that is displayed to the user if the input is empty or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.'

The text will be red when displayed to the user.

5.1.19 ID: Group Creation/Deletion

- Adding a Group
  - Processing:

    If the Group name is empty an error message will be generated and returned and processing stopped. If the Group name is not in the database there is an attempt to parse the room names. A database transaction is started. The string is split along commas and each item is added to the rooms table, duplicate errors are ignored but any other database errors cause an error string to be generated, the transaction rolled back and the message returned. Once the rooms have been added the other inputs are inserted into the Groups table. The transaction is rolled back and an error message is returned if there is already a Group with the given Group name. Then rows are inserted into the G-R table for each room given to associate the group with those rooms. Any errors that come from the database connection will cause the transaction to be rolled back and an error to be returned to the user as well. If the group was added successfully then the transaction is committed and a success message is returned to the user.
  - Data
    - Room list

      A list of string which contain the room names to be associated with this key.
    - Success String

      The string that is displayed to the user if the group was successfully added. It's default value is 'The group has been added to the system.' The text will be black when displayed to the user.
    - Error String

      The string that is displayed to the user if the group name is empty or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

- Removing a Group
  - Processing:

    If the group name is empty, then an error message will be returned and further processing will stop. If the input match the data requirements then there is an attempt to to delete the record from the Groups table instead. This delete will cascade through the P-G and G-R tables. An error message is returned if there is no matching Group name in the database, and any other errors that come from the database connection will be reported to the user as well. If the group was removed successfully then a success message is returned.
  - Data

- Success String

  The string that is displayed to the user if the group was successfully removed. It's default value is 'The group has been removed from the system.' The text will be black when displayed to the user.

- Error String

  The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.20 ID: Remove All Users from Group

- Processing:

  If Group ID is outside of the acceptable range then an error message is returned and no further processing is done. If the input match the data requirements then there is an attempt to remove all record from the P-G table where the group matched Group ID. An error message is displayed if there is no matching Group ID in the database, and any other errors that come from the database connection will be reported to the user as well. If all people were removed from the group successfully then a success message is returned.

- Data

  - Success String

    The string that is displayed to the user if all people were successfully removed from the group. It's default value is 'The people have been removed from that group.' The text will be black when displayed to the user.

  - Error String

    The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.21 ID: Add Person to Group

- Processing:

  If the T-Number is not in the correct form or if the Group ID is outside of the acceptable range then an error message is returned and no further processing is done. If the input match the data requirements then there is an attempt to add the record to the P-G table indicating the person's membership in the group. An error message is displayed if there is no matching T-Number or Group ID in the database, and any other errors that come from the database connection will be reported to the user as well. If the person was added from the group successfully then a success message is returned.

- Data

  - Success String

The string that is displayed to the user if the person was successfully added from the group. It's default value is 'The person has been added from that group.' The text will be black when displayed to the user.

- Error String

  The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.22 ID: Due Date Viewing

- Processing:

  If the date is empty or invalid then an error is returned and processing stops. Otherwise the P-K and all P-$category tables are queried with the date parameter to find all items due back on that date. These items and people will be added to the Due List and returned. Any errors coming from the database connection would cause an error message to be returned.

- Data

  - Due List

    The list of items due back to return. This list will contain the T-Number, first name, and last name of the people who have items due back. It will also have the item category(possibly 'Key') and the Tag Number/Key Name or Descriptor. It is empty by default.

  - Error String

    The string that is displayed to the user if the date is invalid or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.1.23 ID: Key/Item Checkout

- Processing:

  If the T-Number is not in the correct form, the Tag Number is outside of the acceptable range/Descriptor is empty, the person want to check-out more than what is available, or the table indicating which type of item does not exist, then an error message will be returned and further processing will stop. If the input match the data requirements then the user is shown a screen that displays the Borrower Agreement. The agreement consists of, at a minimum, the borrowers T-Number, full name, the items being checked-out, and the due date, if requested. This agreement is to be printed off, signed by the borrower and the user must accept the signed agreement and agree to the check-out. Once that happens there is an attempt to add a record indicating a checkout to either the P-K table or one of the P-$category tables. The amount of inventory available to check-out in either the Keys table or one of the $category tables for that item is also decremented by the amount checked-out. An error message is returned if there is no matching T-Number, Tag Number, or descriptor in the database, and any other errors that come from the database connection will be reported to the user as well. If the inventory was checked-out successfully then a success message is returned.

- Data
  - Borrower Agreement

    The string that is to be printed off and signed by the borrower. This will be made up of, at a minimum, the T-Number and the whole name of the borrower, to be pulled off of the people table, the item being checked-out, the amount of that item to check-out, the due date (if requested), and a place for the borrower to sign in agreement.
  - Success String

    The string that is displayed to the user if the inventory was successfully checked-out. It's default value is 'The check-out was successful.' The text will be black when displayed to the user.
  - Error String

    The string that is displayed to the user if the inputs do not meet the system requirements or when a problem occurs while interacting with the database. It's default value is 'An unknown error occurred.' The text will be red when displayed to the user.

5.2 Data Detailed Design

Note: The processing of the data presented here is completely managed by MySQL and is therefore outside of the scope of this document.

5.2.1 ID: Users Table

- Username
  - Type: varchar
  - Default value: user
  - Use: Unique identifier of a system user
  - Semantics: Identifier of user account
  - Format: Alphanumeric 1 to 20 characters long
  - Acceptable values: Non-null, non-empty, and unique from all other records in the user table.
- Password
  - Type: varbinary
  - Default value: Hash of Password

  - Use: Provide ability for user authentication into the system
  - Semantics: String known only to the user
  - Format: Possibly binary result of hash function
  - Acceptable values: Non-null, and non-empty.

5.2.2 ID: People Table

- T-Number
  - Type: char(9)
  - Default value: 'T00000000'
  - Use: Uniquely identify a person.
  - Semantics: Identification number given to a person by the university
  - Format: It matches to the regular expression T[0-9]{8}
  - Acceptable values: Non-null and non-empty. Must be unique to the People table.
- First Name
  - Type: varchar
  - Default value: FirstName
  - Use: For displaying the person's first name in displays concerning their groups or checked out inventory.
  - Semantics: The person's first name.
  - Format: Alphabetic characters less than or equal to 20 characters in length.
  - Acceptable values: Non-null and non-empty.
- Last Name
  - Type: varchar
  - Default value: LastName
  - Use: For displaying the person's last name in displays concerning their groups or checked out inventory.
  - Semantics: The person's last name.
  - Format: Alphabetic characters less than or equal to 30 characters in length.
  - Acceptable values: Non-null and non-empty.
- Street Address
  - Type: varchar

  - Default value: No address given.
  - Use: For displaying the person's street address in displays concerning their groups or checked out inventory.
  - Semantics: The person's street address.
  - Format: Numeric Street number, followed by alphanumeric street name, possible apartment designation, state name or abbreviation, followed by a numeric Zip code.
  - Acceptable values: Non-null and non-empty.
- Home Phone Number
  - Type: varchar
  - Default value: No home phone given.
  - Use: For displaying the person's home phone number in displays concerning their groups or checked out inventory.
  - Semantics: The persons home phone number.
  - Format: It matches to the regular expression [0-9]{3}-[0-9]{3}-[0-9]{4}
  - Acceptable values: Non-null and non-empty.
- Work Phone Number
  - Type: varchar
  - Default value: No work phone given.
  - Use: For displaying the person's work phone number in displays concerning their groups or checked out inventory.
  - Semantics: The persons work phone number.
  - Format: It matches to the regular expression [0-9]{3}-[0-9]{3}-[0-9]{4}
  - Acceptable values: Non-null and non-empty.
- E-mail address
  - Type: varchar
  - Default value: No e-mail address given.
  - Use: For displaying the person's e-mail address in displays concerning their groups or checked out inventory.
  - Semantics: The persons e-mail address.
  - Format: <string>@<string>
  - Acceptable values: Non-null and non-empty.

5.2.3 ID: Key Table

- Tag Number
  - Type: int
  - Default value: One more than the largest current Tag Number
  - Use: To uniquely identify a set of identical keys in the system.
  - Semantics: The number on the tag that the CSC Department secretary has the physical keys attached to.
  - Format: Positive integer.
  - Acceptable values: 0 – 2147483647. Must be unique to Key table.
- Name
  - Type: varchar
  - Default value: KeyName
  - Use: Identifies a set of identical keys by an alphanumeric designation.
  - Semantics: A sequence of alphanumeric characters engraved on each key.
  - Acceptable values: Any string less than or equal to 30 characters.
- Total number of keys
  - Type: int
  - Default value: 0
  - Use: For displaying the total amount of that set of keys that the CSC Department has control over.
  - Semantics: The total number of that key set that the Department has made.
  - Format: Positive integer.
  - Acceptable values: 0 – 2147483647.
- Amount of Keys available for checkout
  - type: int
  - Default value: 0
  - Use: For displaying the total amount of that set of keys remain in the inventory and available for checkout.
  - Semantics: The number of that set of keys that the CSC Department secretary has in her possession.
  - Format: Positive integer.
  - Acceptable values: 0 – Total number of this key.

5.2.4 ID: P-K Table

- Tag Number
  - Type: int
  - Default value: 0
  - Use: Combines with T-Number to uniquely identify a record.
  - Semantics: The number on the tag that the CSC Department secretary has the physical keys attached to.
  - Format: Positive integer.
  - Acceptable values: 0 – 2147483647. Tag Number, T- Number combination must be unique to P-K table.
- T-Number
  - Type: char(9)
  - Default value: 'T00000000'
  - Use: Combines with Tag Number to uniquely identify a record
  - Semantics: Identification number given to a person by the university
  - Format: It matches to the regular expression T[0-9]{8}
  - Acceptable values: Non-null and non-empty. Tag Number, T- Number combination must be unique to P-K table.
- Number checked out
  - Type: int
  - Default value: 1
  - Use: For display of the number of this type of key checked out by the person
  - Semantics: The number of that type of key checked out by the person.
  - Acceptable values: Positive integer such that this number + the number of this key available for check out <= the total number of this key managed by the CSC department.
- Check out date
  - Type: date
  - Default value: now()
  - Use: For display of the date that the person checked out these keys.
  - Semantics: The date that the person checked out these keys
  - Format: MM/DD/YYYY
  - Acceptable values: 01/01/1000 to the current date.

- Due back date
  - Type: date
  - Default: null
  - Use: If requested, the due back date is used to keep track of when a key or set of keys is due back to the CSC Department
  - Semantics: The data that these keys are due back to the department
  - Format: MM/DD/YYYY or null
  - Acceptable: A date in the future, or null if a due date is not requested.

5.2.5 ID: Groups Table

- Group ID
  - Type: int
  - Default: One more than the largest current Group ID.
  - Use: A unique identifier for a group.
  - Semantics: A unique identifier for a group.
  - Format: Positive Integer.
  - Acceptable values: 0 – 2147483647. Must be unique to Groups table.
- Group Name
  - Type: varchar
  - Default value: GroupName
  - Use: Identifies a group by an alphabetic designation.
  - Semantics: The groups name
  - Acceptable values: Any string less than or equal to 30 characters.

5.2.6 ID: P-G Table

- Group ID
  - Type: int
  - Default: 0.
  - Use: Combines with T-Number to uniquely identify a record
  - Semantics: A unique identifier for a group.
  - Format: Positive Integer.
  - Acceptable values: 0 – 2147483647. Group ID, T- Number combination must be unique to

P-G table.

- T-Number
  - Type: char(9)
  - Default value: 'T00000000'
  - Use: Combines with Group ID to uniquely identify a record
  - Semantics: Identification number given to a person by the university
  - Format: It matches to the regular expression T[0-9]{8}
  - Acceptable values: Non-null and non-empty. Group ID, T- Number combination must be unique to P-G table.

5.2.7 ID: Rooms Table

- Room ID
  - Type: int
  - Default: One more than the largest current Room ID.
  - Use: A unique identifier for a room.
  - Semantics: A unique identifier for a room.
  - Format: Positive Integer.
  - Acceptable values: 0 – 2147483647. Must be unique to Rooms table.
- Room Name
  - Type: varchar
  - Default value: RoomName
  - Use: Identifies a room by an alphabetic designation.
  - Semantics: The rooms name.
  - Acceptable values: Any string less than or equal to 30 characters.

5.2.8 ID: K-R Table

- Tag Number
  - Type: int
  - Default value: 0
  - Use: Combines with Room ID to uniquely identify a record.
  - Semantics: The number on the tag that the CSC Department secretary has the physical keys attached to.

  - Format: Positive integer.
  - Acceptable values: 0 – 2147483647. Tag Number, Room ID combination must be unique to K-R table.
- Room ID
  - Type: int
  - Default: 0
  - Use: Combines with Tag Number to uniquely identify a record.
  - Semantics: A unique identifier for a room.
  - Format: Positive Integer.
  - Acceptable values: 0 – 2147483647. Tag Number, Room ID combination must be unique to K-R table.

5.2.9 ID: G-R Table

- Group ID
  - Type: int
  - Default: 0.
  - Use: Combines with Room ID to uniquely identify a record
  - Semantics: A unique identifier for a group.
  - Format: Positive Integer.
  - Acceptable values: 0 – 2147483647. Group ID, Room ID combination must be unique to G-R table.
- Room ID
  - Type: int
  - Default: 0
  - Use: Combines with Group ID to uniquely identify a record.
  - Semantics: A unique identifier for a room.
  - Format: Positive Integer.
  - Acceptable values: 0 – 2147483647. Group ID, Room ID combination must be unique to G-R table.

Note: The following two tables will be created for each new category of inventory that will be created. I will use $category as a stand in for the category name.

5.2.10 $category Table

- Designator
  - Type: varchar
  - Default value: ItemID
  - Use: To uniquely identify a set of items in $category.
  - Semantics: A unique identifier for the item
  - Format: Alphanumeric of up to 50 characters.
  - Acceptable Non-null, non empty, and unique to the $category table.
- Description
  - Type: varchar
  - Default value: ItemDescription
  - Use: For the display of the description of the item.
  - Semantics: A description of the item
  - Acceptable values: Any string less than or equal to 255 characters.
- Total number of items
  - Type: int
  - Default value: 0
  - Use: For displaying the total amount of that set of items that the CSC Department has control over.
  - Semantics: The total number of that item that the Department has.
  - Format: Positive integer.
  - Acceptable values: 0 – 2147483647.
- Amount of items available for checkout
  - type: int
  - Default value: 0
  - Use: For displaying the total amount of that set of items remain in the inventory and available for checkout.
  - Semantics: The number of that item that the CSC Department secretary has in her possession.
  - Format: Positive integer.

    - Acceptable values: 0 – Total number of this item.

5.2.11 P-$category

- Type: Database Table
- Purpose: To store information concerning which items in $category have been checked out by which people.
- Function: Contains information on item checkouts. This includes: the designator for the item, the Tnumber of the person, the number of that item checked out, date the item was checked out, and date the item is due back (if requested).
- Subordinates: None
- Designator
    - Type: varchar
    - Default value: ItemID
    - Use: Combines with T-Number to form a unique record.
    - Semantics: A unique identifier for the item
    - Format: Alphanumeric of up to 50 characters.
    - Acceptable values: Non-null and non-empty. Designator, T- Number combination must be unique to P-$category table.
- T-Number
    - Type: char(9)
    - Default value: 'T00000000'
    - Use: Combines with Designator to uniquely identify a record
    - Semantics: Identification number given to a person by the university
    - Format: It matches to the regular expression T[0-9]{8}
    - Acceptable values: Non-null and non-empty. Designator, T- Number combination must be unique to P-$category table.
- Number checked out
    - Type: int
    - Default value: 1
    - Use: For display of the number of this item checked out by the person
    - Semantics: The number of that item checked out by the person.
    - Acceptable values: Positive integer such that this number + the number of this item

available for check out <= the total number of this item managed by the CSC department.

- Check out date
    - Type: date
    - Default value: now()
    - Use: For display of the date that the person checked out these items.
    - Semantics: The date that the person checked out these items.
    - Format: MM/DD/YYYY
    - Acceptable values: 01/01/1000 to the current date.
- Due back date
    - Type: date
    - Default: null
    - Use: If requested, the due back date is used to keep track of when an item or set of itemss is due back to the CSC Department
    - Semantics: The data that these itemss are due back to the department.
    - Format: MM/DD/YYYY or null
    - Acceptable: A date in the future, or null if a due date is not requested.