|  |
| --- |
| MC214 – SYSTEM ANALYSIS & DESIGN SEM1 MAJOR PROJECT |
| DWU Accommodation System |
| Stafford, Daniel & Louis |
|  |
|  |
|  |

Contents

[Brief Project Information 2](#_Toc421298694)

[Requirements Engineering 2](#_Toc421298695)

[Use Case Diagram 2](#_Toc421298696)

[Use case full description 4](#_Toc421298697)

[Activity Diagram 13](#_Toc421298698)

[Domain Class Diagram 21](#_Toc421298699)

[System Sequence Diagram 22](#_Toc421298700)

[Design Models 26](#_Toc421298701)

[Design Class Diagram 26](#_Toc421298702)

[Sequence Diagram 28](#_Toc421298703)

[Component Diagram 31](#_Toc421298704)

[Package Diagram 32](#_Toc421298705)

[ERD Diagram 33](#_Toc421298706)

# Brief Project Information

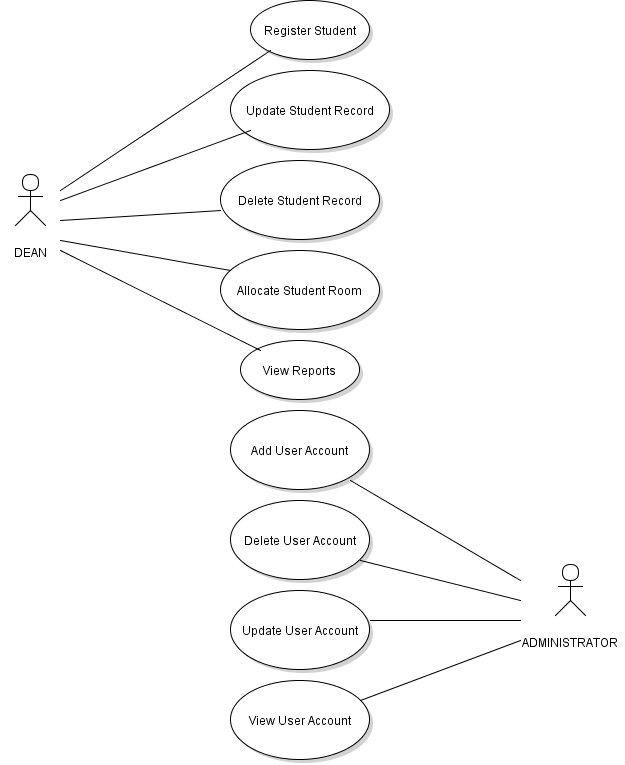
|  |  |
| --- | --- |
| Project Name | DWU Student Accommodation System |
| Developers | Stafford Koki, Daniel Nelson, Louis Ronald |
| Programming Language | Python version 3.4 |
| Database Platform | MySQL |
| Project Description | The DWU Student Accommodation is the proposed system, or, an improvement of the existing accommodation system that will provide improved management of student accommodation information. |

# Requirements Engineering

## Use Case Diagram

The following presents typical uses of the proposed DWU Student Accommodation System:

|  |  |
| --- | --- |
| **Use Case** | **Description** |
| Register student | This use case involves the intended user of the system ‘registering’ or entering the details of a student’s accommodation status into the system. |
| Update Student Information | For this case, it involves making changes to student accommodation information particularly when a student changes rooms, is suspended, is terminated or for any other factors that may involve tempering with student accommodation information. |
| Delete Student Record | As its name suggests, this use case involves unregistering or deleting a student’s accommodation record from the system. |
| View Reports & Queries | For this use case, the intended system user is able to make queries and or create standard, professional looking reports based on his/her search criterion. |
| Archive accommodation data | Archiving is a case where the intended system user may wish to perform a backup of student accommodation data for future reference if necessary. |
| Add user account | The administrator adds a new user for the database with a specified username and password. |
| Delete user account | The administrator deletes an existing user account to the database. |
| Update user account information | The administrator updates user account information |



## Use case full description

The following tables describes in more detail the different use cases in the proposed DWU accommodation system as stated above:

|  |  |  |
| --- | --- | --- |
| Use case name | Register a student | |
| Scenario | A student has paid all necessary fees but most importantly the registration fee and needs to be offered an accommodation on-campus. | |
| Triggering event | Student has paid registration fee and needs to be accommodated in on-campus accommodation at DWU. | |
| Brief description | During the process, a student is allocated a room after being registered or recorded in the DWU accommodation system. Furthermore, the dean issues the student a key to the room in which case a student is legally able to occupy the room. | |
| Actors | Dean (of men or women accordingly) | |
| Related use cases | None | |
| Stakeholders | DWU finance, DWU admin | |
| Preconditions | The student has paid registration fee and the necessary on-campus accommodation fees and presents receipt to dean. | |
| Post-conditions | Student has been recorded on the system, and has been assigned a room on-campus at DWU.  Student gets to be issued a key to his/her room. | |
| Flow of activities | **Actor**   1. Student visits the respective (male or female) dean’s office and presents recipient of registration fee payment 2. Respective dean enters student basic information such as first name, last name, course, faculty, etc. and assigns student to a room. 3. Dean issues a key to the student for him/her to access the assigned room. | **System**   1. System prompts user for basic student information such as first name, last name, course, faculty, year level, etc. 2. System prompts dean for a particular accommodation type 3. System displays all empty rooms for that particular accommodation type and prompts user to input the room number. 4. System validates whether or not room is available and notifies user accordingly. 5. System creates new student accommodation account and stores all the prompted data from user including the accommodation and room information of the student. |
| Exception conditions | All the rooms have been fully booked.  User has not entered all of the required student information as prompted by the system.  Student has not registered. | |

|  |  |  |
| --- | --- | --- |
| Use case name | Update student accommodation information | |
| Scenario | Student needs to change his/her information in the proposed DWU accommodation system or even change his/her room. | |
| Triggering event | Student presents legal civil registry documents as evidence of change of name or other personal information.  Student expresses his/her desire to change his/her room based on a valid reason. | |
| Brief description | Involves the dean aiding the student in changing his/her information which are stored on the DWU accommodation system. The type of information to change includes the student’s personal information such as first name and last name etc., and the accommodation information such as the room number etc. | |
| Actors | Dean (of men or women accordingly) | |
| Related use cases | None | |
| Stakeholders | DWU Finance, DWU admin | |
| Preconditions | Student must have a valid reason for the change of rooms or, must have legal documents that provide evidence of a student’s change of personal information such as first name, last name etc. | |
| Post-conditions | Updated student information is stored by the system  Student gets issued the new room key, transfers possessions from old room to new room, then returns the old key to the respective dean. | |
| Flow of activities | **Actor**   1. Dean validates and verifies the student’s reason for the change of room or student information. 2. Upon approval, dean gathers the requested information that need to be changed. 3. Dean searches for student accommodation record preferably by ID number. 4. Dean edits the existing fields in the student information page with those requested for change by student. 5. Dean saves the changes 6. Dean notifies the student of the status of the change ultimately. | **System**   1. System prompts user for optional information to perform a search (that is, first name, last name, id number, course code, etc.) 2. System finds students based on search query parameters and displays it to user. 3. System displays all the information related to the particular student record specified by the user. 4. System unlocks all fields and enables user to make edits of the data in the fields 5. Upon user click of ‘update information’ or something alike, system validates and or verifies the changes made by user and notifies dean accordingly for a successful or unsuccessful information entry. 6. Upon successful information entry or update, system saves the changes to the database. 7. System notifies user that the changes to that particular student record has been saved successfully. |
| Exception conditions | Student has no accommodation record in the database. | |

|  |  |  |
| --- | --- | --- |
| Use case name | Delete student record | |
| Scenario | Student is suspended, terminated, or may even have his boarding privileges withheld. | |
| Triggering event | Student commits an offence either physical or academic. | |
| Brief description | In this use case, a student’s record on the accommodation system is deleted or removed permanently. | |
| Actors | Dean (of men or women accordingly) | |
| Related use cases | None | |
| Stakeholders | DWU Finance, DWU admin | |
| Preconditions | Dean has received confirmation from the administration, finance, or any other superior that requests removing a student’s accommodation privilege. | |
| Post-conditions | Student vacates the room  Student returns key to administration (particularly accommodation registry office). | |
| Flow of activities | **Actor**   1. Dean receives confirmation for a particular student’s accommodation record deletion. 2. Dean places a search on the particular student. 3. Dean explicitly deletes the student record from the system using the ‘delete’ button or something alike. 4. Dean returns confirmation to superior of the student accommodation record deletion. | **System**   1. System prompts the user for the search query parameters. 2. System searches the database for student records based on the specified parameters. 3. System provides functionality/feature for record deletion to user. 4. Upon user selection, System deletes the student accommodation record as specified by the user. 5. System notifies user that the specified record has been deleted successfully. |
| Exception conditions | Student has not accommodation record in the system. | |

|  |  |  |
| --- | --- | --- |
| Use case name | View reports & queries | |
| Scenario | The user would like to view all the students, rooms, or accommodations that meet specific requirements. | |
| Triggering event | The user gets request from stakeholders that require specific accommodation information. Or, the dean would like to perform some administrative work on the accommodation based on reports from the system. | |
| Brief description | This use case involves creating aesthetically sound reports that may be displayed on-screen, or may be turned into ‘pdf’ documents. | |
| Actors | Dean (of men or women respectively) | |
| Related use cases | None | |
| Stakeholders | DWU Administration, Registrar, ICT Services, DWU Finance. | |
| Preconditions | User must specify a criteria of search for queries and reports  User must enter a descriptive title of the search results for reports  User must enter a description of the search results for reports | |
| Post-conditions | Results of query/report must be displayed on-screen or written to ‘pdf’ accordingly, based on user preference. | |
| Flow of activities | **Actor**   1. The dean first specifies the table in which he/she would like to specify a search. 2. The dean or the respective actor specifies a certain criteria of search. 3. The actor then commands for the search to begin through a GUI button. 4. If search results exist, results are displayed on the screen in a tabular form. 5. Actor is given the option to save the results as a ‘pdf’ file. | **System**   1. System prompts the user to specify the table in which to perform the search. 2. System then prompts the user to specify the criteria of search. 3. As commanded by the actor, the system performs the search criteria on the database. 4. Whether or not results exist in the database, the system displays the information as necessary. 5. System gives the option for the actor to save the information in a ‘pdf’ format preferably. |
| Exception conditions | There is no data and or records in the system database | |

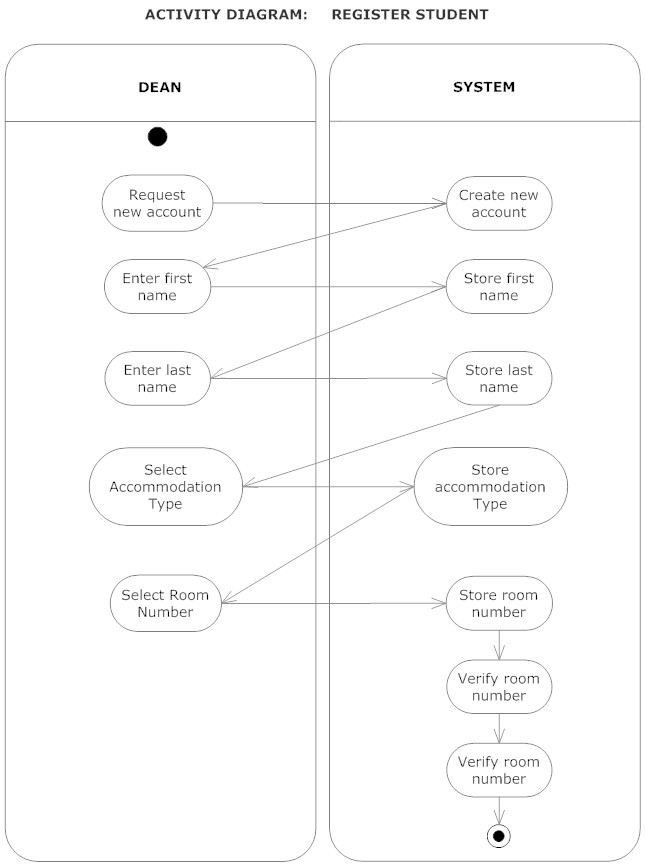
|  |  |  |
| --- | --- | --- |
| Use case name | Add user account | |
| Scenario | Administrator would like to add a new user account to the system, enabling him/her to manipulate the database in ways restricted by his/her security level privileges. | |
| Triggering event | A new user needs to access to the system database. | |
| Brief description | This involves creating a new user account that enables another person to access the system and manipulate data. The actions or activities that may be performed by this user are restricted by his/her security level. | |
| Actors | System Administrator (ICT) | |
| Related use cases | None | |
| Stakeholders | DWU Administration | |
| Preconditions | New user account to be created must not already exist  The particular user must not already have another account. | |
| Post-conditions | User login credentials must then be provided to the user by the administrator. | |
| Flow of activities | **Actor**   1. System administrator enters basic personal information of the user such as first name, last name and gender 2. System administrator then enters the username and temporary password of the particular user. 3. System administrator then saves the user account record in the database. | **System**   1. System prompts the administrator to enter the user’s personal information particularly first name, last name, and the gender 2. System then prompts the user for the username and temporary password of the user. 3. After user entry, system confirms that there is no other account in the database with the same username. 4. Upon confirmation that there is no other account similar to the one that’s being created, system saves user account record to database. 5. Otherwise, system notifies the user that there is already an account with the same username. Thus, the system repeats process beginning from step (2). |
| Exception conditions | The particular user already has an account with the system. | |

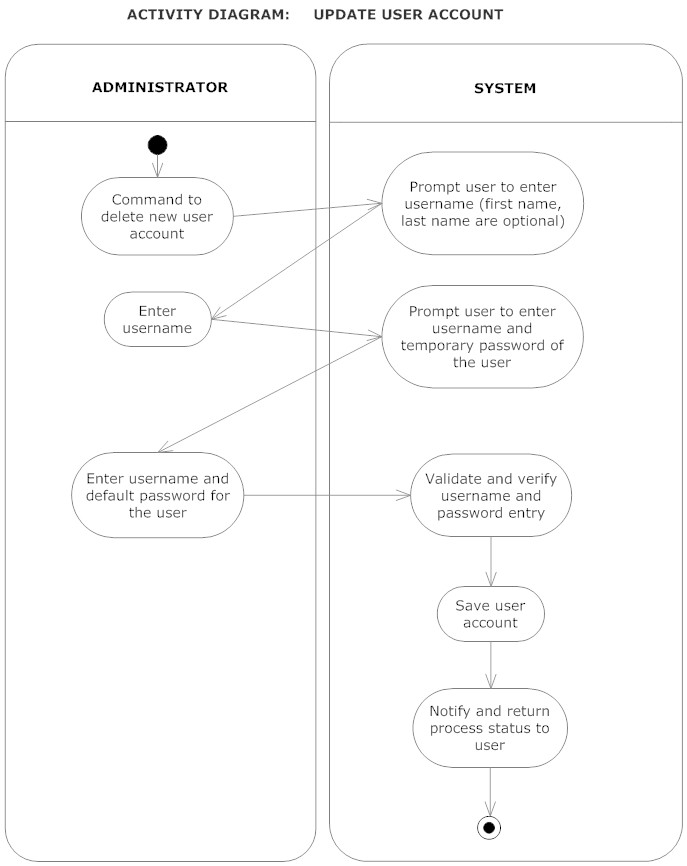
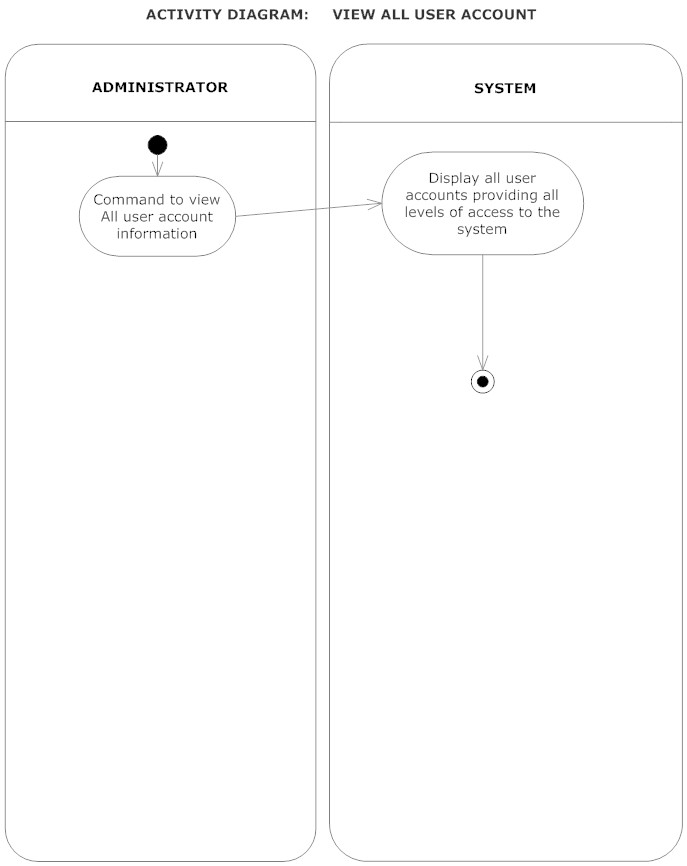
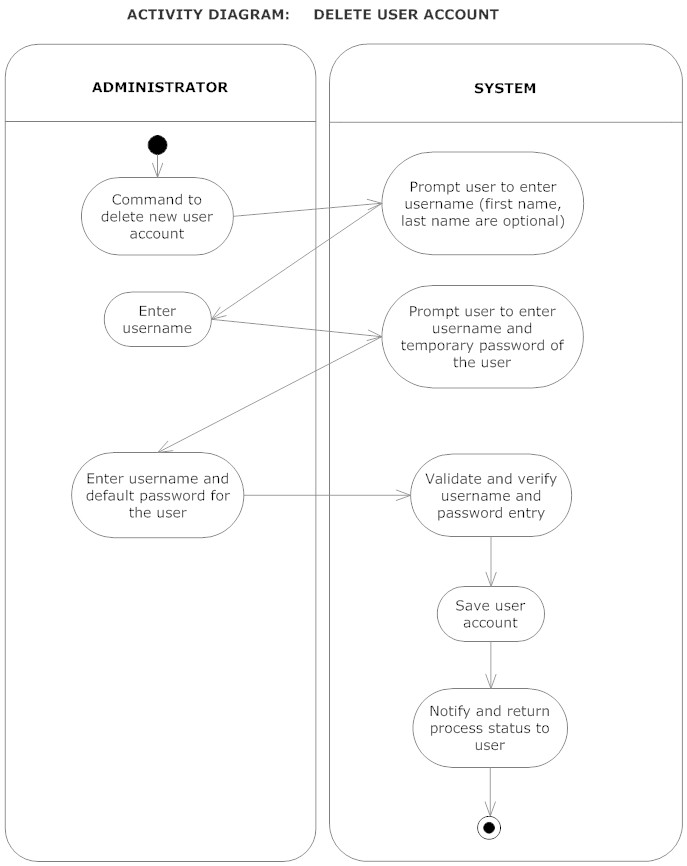
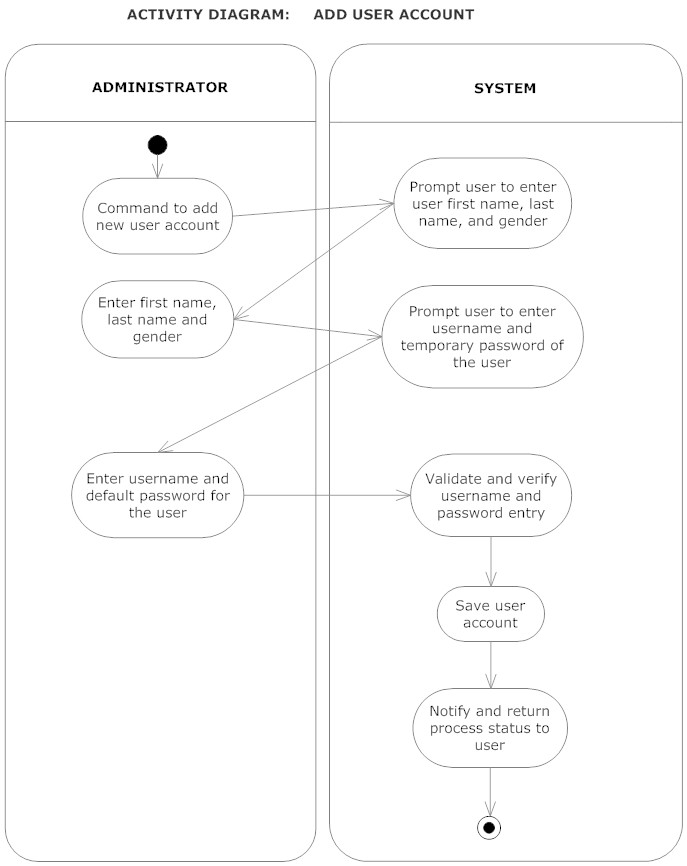
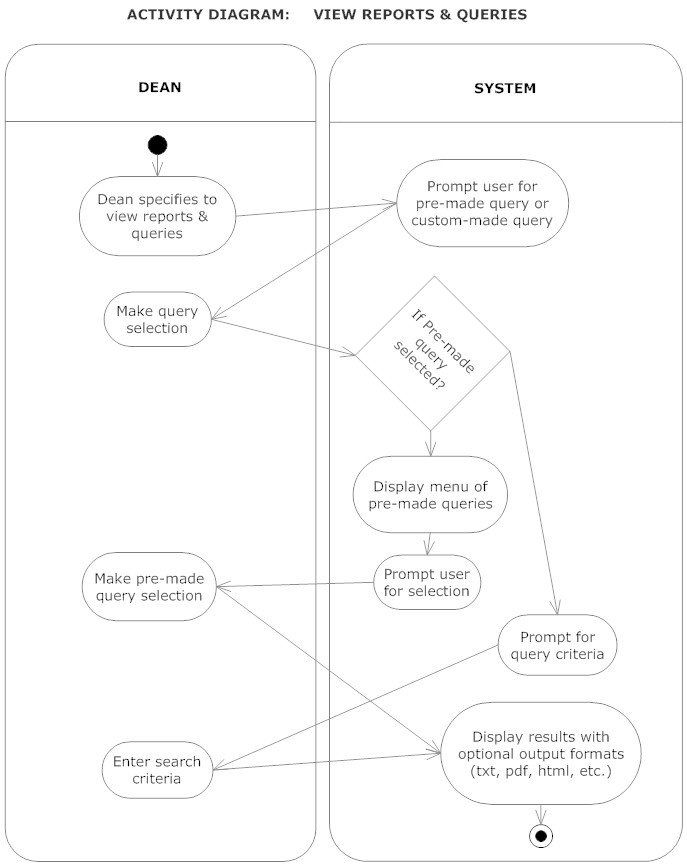
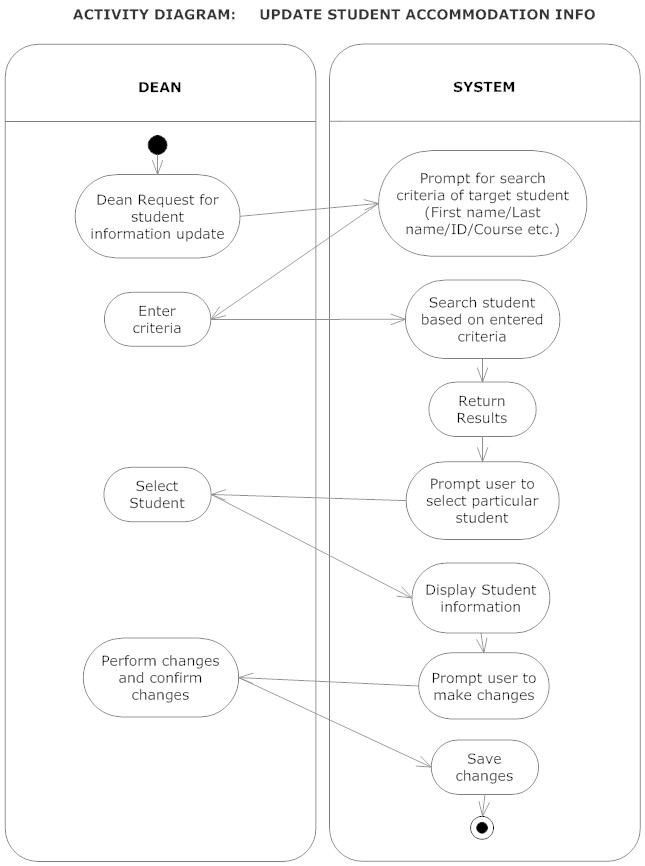
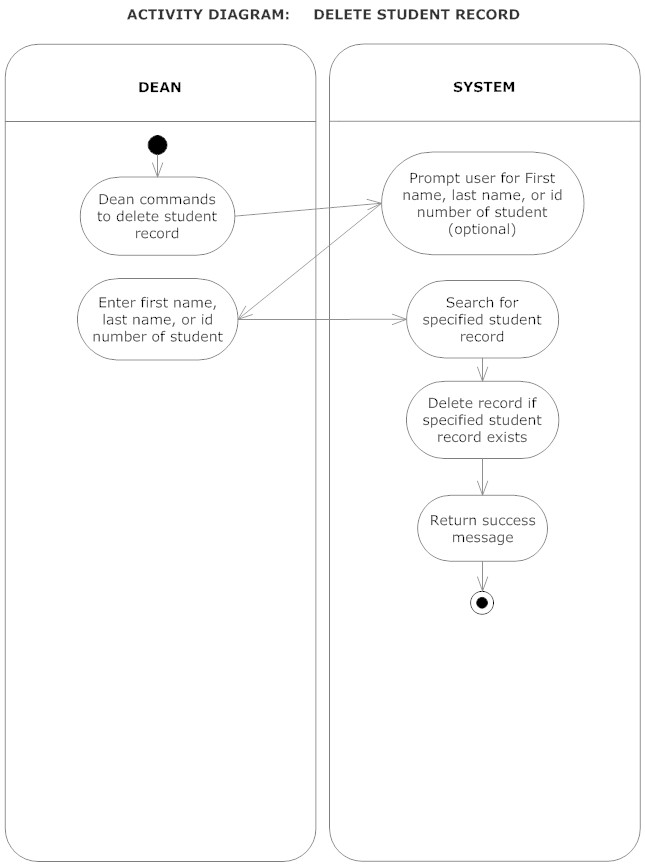
|  |  |  |
| --- | --- | --- |
| Use case name | Delete User Account | |
| Scenario | Administrator would like to delete a user account in the system, disabling him/her to manipulate the database in ways restricted by his/her security level privileges. | |
| Triggering event | An existing user needs to be deleted and have his/her access to the system database removed. | |
| Brief description | This involves deleting user account that enables another person to access the system and manipulate data. | |
| Actors | System Administrator (ICT) | |
| Related use cases | None | |
| Stakeholders | DWU Administration | |
| Preconditions | New user account to be created must not already exist  The particular user must not already have another account. | |
| Post-conditions | User login credentials must then be provided to the user by the administrator. | |
| Flow of activities | **Actor**   1. System administrator enters username of the user account to be deleted (first name and last name fields are optional if the username is forgotten) 2. System administrator specifies to delete the selected (and or currently opened) user account. | **System**   1. System prompts the administrator to enter username 2. System searches for the particular user account by the username and displays the result 3. System prompts the user for various options including to delete user account. 4. Upon user selection to delete the displayed account, the system deletes the user account permanently. 5. System displays status of process to user. |
| Exception conditions | The particular user already has an account with the system. | |

|  |  |  |
| --- | --- | --- |
| Use case name | Update user account information | |
| Scenario | Administrator would like full access to a user account enabling him/her to change user account personal information such as first name and last name, or even user login credentials such as username and password. | |
| Triggering event | User account information related to a particular user needs to be updated as a result of: name change at civil registry, security measures conducted to change user passwords, deactivating/suspending a user’s account as per administrative instructions. | |
| Brief description | This use case involves the following: changing of user account personal information such as first name and last name; changing of user login credentials such as username and password; deactivating of user accounts; | |
| Actors | System Administrator (ICT) | |
| Related use cases | None | |
| Stakeholders | DWU Administration | |
| Preconditions | The user account to be updated must already exist | |
| Post-conditions | The user must be made aware of the changes that have been made on his/her login account. | |
| Flow of activities | **Actor**   1. Administrator enters his/her login credentials to have special administrative access to the system. 2. Administrator performs a search for the particular account to be manipulated. 3. Administrator then makes the changes to the field data belonging to that particular user account. | **System**   1. System prompts user to enter criteria of search for the particular user account he/she would like to edit 2. System searches through the database for specified user account. 3. If user account exists in the database, system displays the full user account information to the user, and makes fields editable for the user to make changes. 4. Otherwise, system displays to the user that the specified user account cannot be found in the database or, is simply non-existent. |
| Exception conditions |  | |

|  |  |  |
| --- | --- | --- |
| Use case name | View all User Accounts | |
| Scenario | Administrator would like to view all the login user accounts that are registered in the system. | |
| Triggering event | Administrator would like a list of all the user accounts registered under the system for technical applications. | |
| Brief description | This particular use case involves the system administrator viewing all the user accounts that are registered under the system. | |
| Actors | System Administrator (ICT) | |
| Related use cases | None | |
| Stakeholders | DWU Administration, DWU ICT Support | |
| Preconditions | User accounts must exist on the system database. | |
| Post-conditions | The user must be made aware of the changes that have been made on his/her login account. | |
| Flow of activities | **Actor**   1. User enters his/her account credentials when logging in to the system. 2. Upon successful authentication, user (administrator) is granted access to the system. 3. User selects to view all the login user accounts credentials that are registered or stored in the system’s database. | **System**   1. System prompts user for user login credentials. 2. Upon entry, system verifies and validates the entered user credentials for its existence. 3. Assuming all is well, the user (administrator) is granted access to the system. 4. Upon user selection to view all the user accounts recorded on the system database, the system displays a list of all the user login credentials and other accompanying information in a tabular form. |
| Exception conditions |  | |

## Activity Diagram

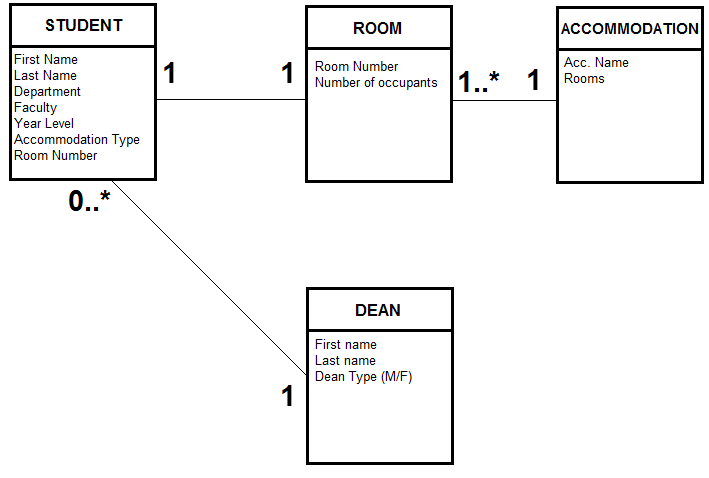




## Domain Class Diagram

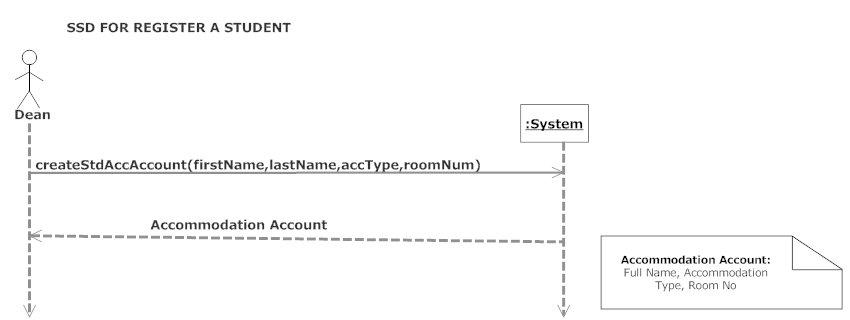
In the problem space, the classes and or objects are identified as:

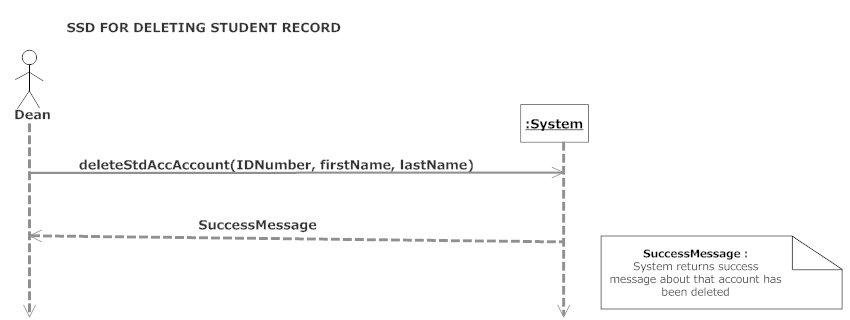
|  |  |  |
| --- | --- | --- |
| **Class** | **Attributes** | **Description** |
| Student | First name  Last name  Department  Faculty  Year level  Accommodation type  Room number | This is the recipient of an accommodation service at DWU. A student gets an allocated room by the dean in a specific type of accommodation on-campus. |
| Accommodation | Name  Number of rooms | This represents the type of accommodation that is provided at DWU which varies such as units, halls, cottages, etc. |
| Room | Room number  Number of occupants | Represents a room within the accommodation type that accommodates a number of students. |
| Dean | First name  Last name  Dean type (Female or male dean) | The dean is the collective term for either male or female, and aids the student in the process of settling in for accommodation at Divine Word University. |

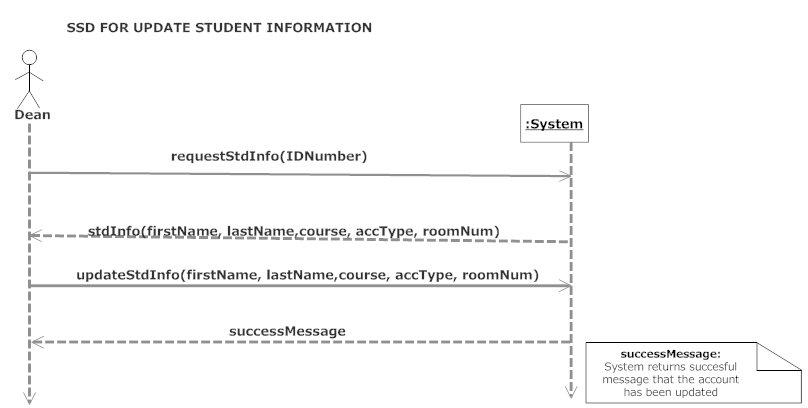


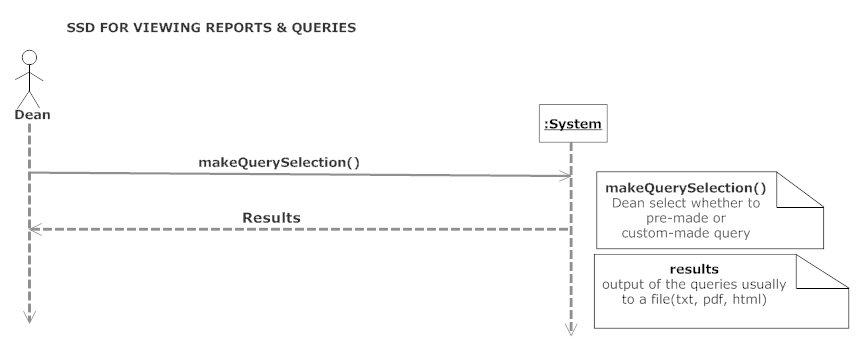
## System Sequence Diagram

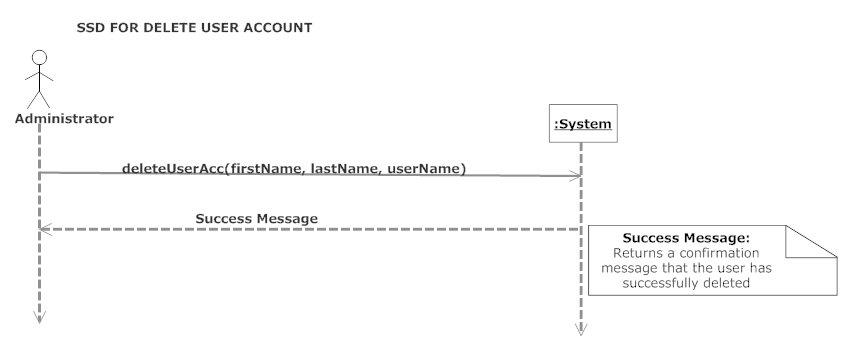
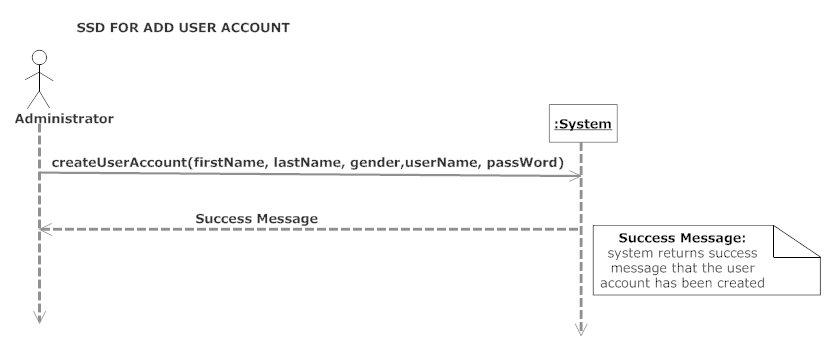
The following diagrams basically demonstrate system sequence diagrams for each of the use cases identified earlier. For each use case, the corresponding SSD diagrammatically explains the interaction between the system and the specific user.

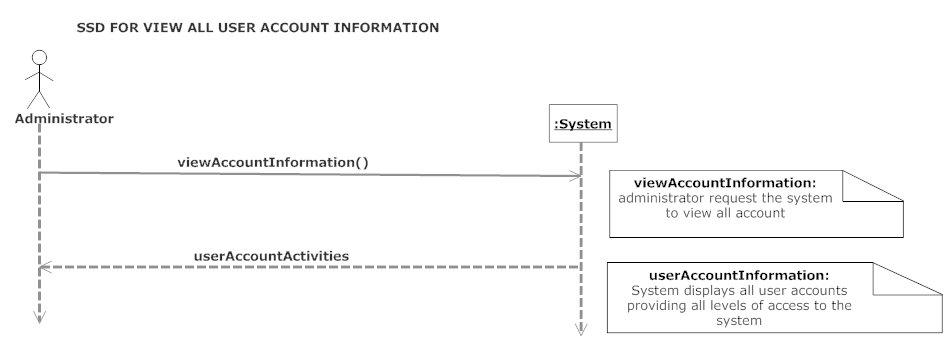


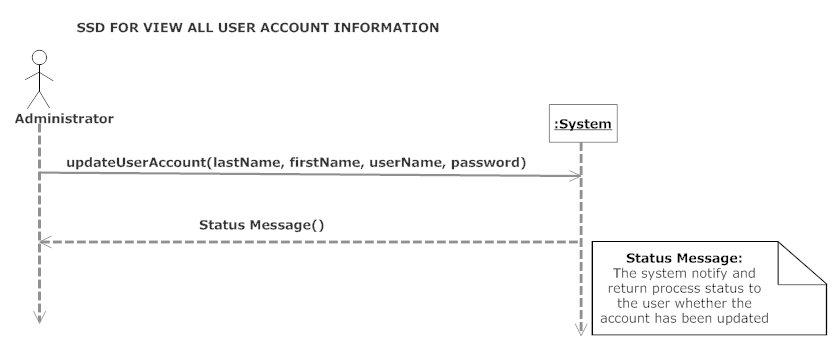






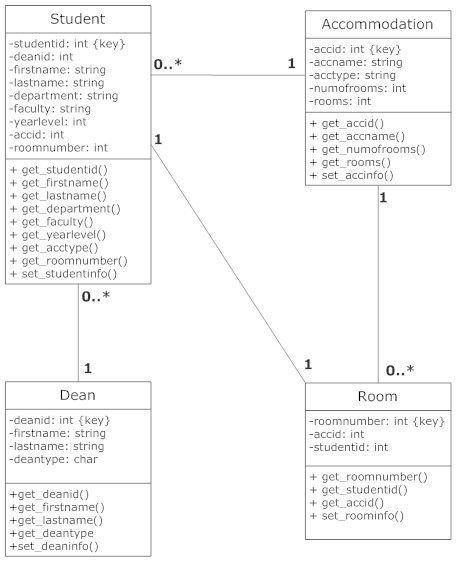


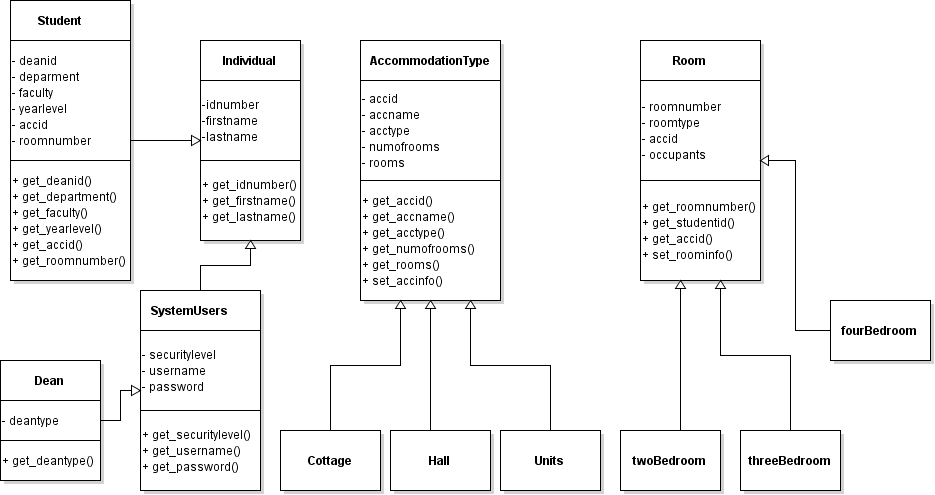




# Design Models

## Design Class Diagram





## Sequence Diagram

**1. Register a Student**

**Dean**

**System**

**Success Message:** System return success message about the account has been deleted

**Success Message**

**DeleteStdAccomAccount**

**(StdIDNumber, FrstName, LstName)**

**2. Deleting Student Record**

**System**

**Dean**

**Accommodations Account**

**CreateStdAccomAccount**

**(firstname, lastname, accomtype, roomnumber)**

**Accommodation Type:**

FullName, Accommodation Type, room number

**3. Update Student Information**

**Dean**

**System**

**RequestStdInfo (IDNumber)**

**StdInfo (firstname, lastname, course, accomtype, roomnumber)**

**Success Message:** System returns successful message that the account has been updated

**Success Message**

**UpdateStdInfo (firstname, lastname, course, accomtype, roomnumber)**

4. **Viewing Reports and Query**

**System**

**Dean**

**Results:** output of the queries usually to a file (text, pdf, html)

**Make Query Selection ():**

Dean select whether to pre-made or custom-made query

**Result**

**MakeQuerySelection ()**

**5. Add User Account**

**Administrator**

**Success Message:** System returns success message that the user account has been created

**Success Message**

**CreateUserAccount (firstname, lastname, gender, userName, passWord)**

**System**

**6. Delete User Account**

**Success Message:** Return confirmation message that the user has successful deleted

Success Message

DeleteUserAcc (firstName, lastName, userName)

**Administrator**

**System**

**7. View all User Account Information**

**UserAccountInformation:**

System displays all user accounts providing all levels of access to the system

**ViewAccountInformation:**

Administrator request the system to view all account.t

**UserAccountActivities**

**ViewAccountInformation ()**

**System**

**Administrator**

**8. View all User Account Information**

**System**

**Administrator**

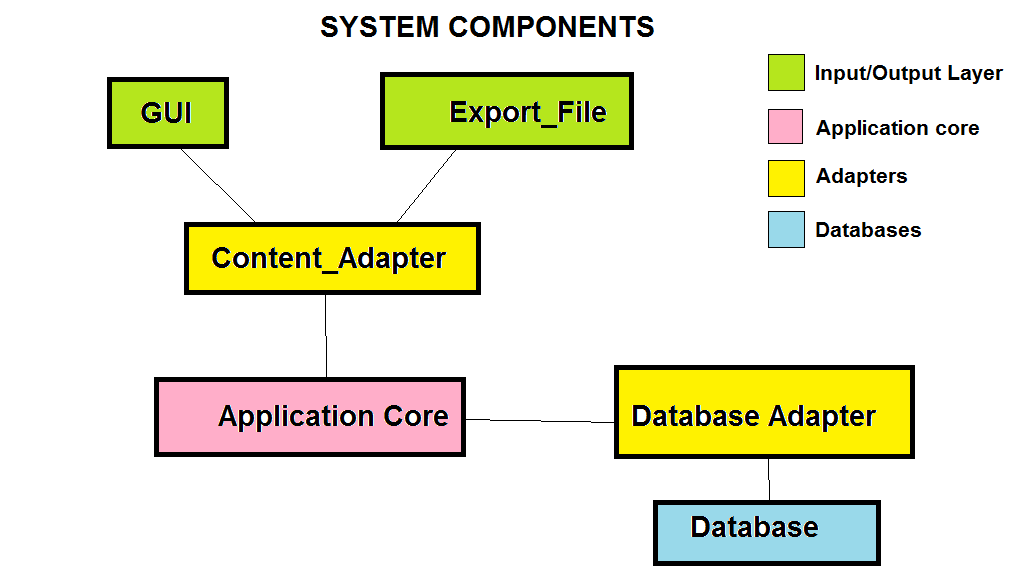
**Status Message:**

System notifies and returns process status to the user whether the account has been updated

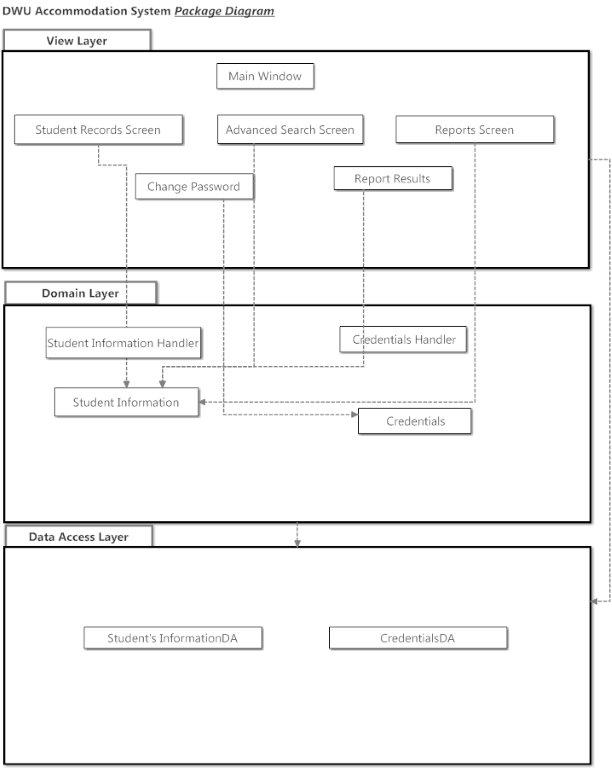
**Status Message ()**

**UpdateUserAccount (firstName, lastName, userName, password)**

## Component Diagram



## Package Diagram



## ERD Diagram

