Software Requirements Specification

for

Centennial Residence Place Portal

Version 1.2 approved

Prepared by COMP225 Group 3

Centennial College

Aug 5, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Kam Hung Chan	18 May 2022	Update 1.4 & 2.1	1.0
Yuk Sing Cheung	18 May 2022	Update 2.2, 2.3 & 2.4	1.0
Nanjin Wang	18 May 2022	Update 3.1 & 3.2	1.0
Ziyan Liu	19 May 2022	Update 1.1,1.2&1.3	1.0
Jinsung Ahn	19 May 2022	Update 2.5, 2.6 &2.7	1.0
Nanjin Wang	19 May 2022	Update 3.3 & 3.4	1.0
Yuk Sing Cheung	21 May 2022	Update 2.2, 2.3, Appendix C (Stakeholder register, interview questions)	1.0
Kam Hung Chan	22 May 2022	Update 1.4 & 2.1	1.0
Ziyan Liu	26 May 2022	Update 1.1,1.3 & interview Q&A (3,4,5)	1.0
Kam Hung Chan	31 May 2022	Update 4.1	1.0
Jinsung Ahn	31 May 2022	Update 2.5, 2.6 &2.7	1.0
Nanjin Wang	31 May 2022	Update 3.4	1.0
Yuk Sing Cheung	31 May 2022	Update 5.1	1.0
Nanjin Wang	4 June 2022	Update 5.2&5.3	1.0
Jinsung Ahn	4 June 2022	Update 2.5, 2.6 ,2.7 & interview Q&A	1.0
Kam Hung Chan	4 June 2022	Update 1.5, 2.1 & Appendix C (interview questions)	1.0
Yuk Sing Cheung	4 June 2022	Update 2.2 & Appendix C (interview questions)	1.0
Ziyan Liu	5 June 2022	Update 4.1.1 &5.5 Appendix A	1.0
Yuk Sing Cheung	16 June 2022	Update 3.5 (Use case table)	1.1

Kam Hung Chan	16 June 2022	Update 3.5 (Use case table)	1.1
Ziyan Liu	16 June 2022	Update 4 (Use case diagram FR02 & FR03)	1.1
Nanjin Wang	19 June 2022	Update 4 (4.1.1)	1.1
Kam Hung Chan	19 June 2022	Update 1.1 - 1.3, 2.2, 3.1, 3.2, 4.1, 5.5 & Appendix C	1.1
Yuk Sing Cheung	19 June 2022	Update 4 (4.1.1)	1.1
Ziyan Liu	20 June 2022	Update 4.1 part 1 & 4.1.2	1.1
Jinsung Ahn	22 June 2022	Update 4.1.1 & 4.1.2	1.1
Kam Hung Chan	23 June 2022	Update 4.1 fig.3 use case diagram	1.1
Yuk Sing Cheung Nanjin Wang Kam Hung Chan Ziyan Liu Jinsung Ahn	16 July 2022	Appendix E (Domain class diagram, CRC index card)	1.1
Yuk Sing Cheung	26 July 2022	Appendix E (State diagram)	1.2
Nanjin Wang	26 July 2022	Appendix E (Sequence Diagram)	1.2
Yuk Sing Cheung	29 July 2022	Updated the Part B of the project	1.2
Kam Hung Chan	29 July 2022	Updated the Part B of the project	1.2
Ziyan Liu	29 July 2022	Appendix E (State diagram)	1.2
Kam Hung Chan	31 July 2022	Appendix E (Class diagram – Party Analysis Pattern) and 1.4 References	1.2
Jinsung Ahn	3 August 2022	technical review for Group2 Part A & Part B // Deliverable #3	1.2
Yuk Sing Cheung	5 Aug 2022	Appendix E (State diagram)	1.2

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

This new software “Centennial Residence Place Portal” addresses the problem of students mismatching a roommate who has different living style and value. The app will help new coming students to match their roommates who can share the same values, culture and interests. This is version 1, and the app will connect to external applications and management systems.

For further development of this application may consider adding rent payments system and enquiries system. Another SRS for payment system and enquiries system will be needed apart from this SRS for room matching system.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

Acronyms	Description
Student(s)	Student who are residents in the Residence Place
SRS	Software Requirements Specification
SQL	Sequential query language
DB	Database

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

The intended audience for this project is developers, project managers, users (including students and lease manager), and testers. The software may improve incrementally with new features. It is suggested to read through the document from part. 1 to part.6 with user the appendixes supplementary.

For students, new students who are from overseas may not be familiar with English, thus simple English is suggested throughout this project. They also concern about the features of the application as well as their privacy issue while they are using the application.

For lease managers: they are working in the property management company named “American Campus Communities”. While they are facing that so many students having disputes with their roommates and increasing workload in room arrangement, they are looking forward that this app will provide huge help in room arrangement and reduce the workload in settling disputes and room re-arrangement.

For administrators: administrators are the administrators of the portal, and they are also the front desk and the administrators of the Centennial Residence Place.

For developers, project manager and tester: this document is prepared by the project team and approved by the project manager. This document should be written in a consistent and traceable way, so that the developer, project manager and tester can go through it, understand it, discuss it and work for it.

Product Scope
<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

Many students who live in Centennial Residence Place have randomly paired up their roommates before they move in. It leads to many disputes among roommates with different personalities, habits and lifestyles. “Centennial Residence Place Portal” is a web and mobile application that provides a channel for residents to know each other and to find the right roommate. By using this portal, residents can firstly register and create their own profiles. Then, they can initiate a post to seek for their roommates by further introducing themselves and setting general rules for the shared rooms. Residents may also browse all the pairing posts to find potential roommates. Once residents find someone good for pairing up, they can have further discussion and conversation via chat box, and they can send pair-up requests after that. Once the requests approved by the initiators and leasing managers, and then they can be roommates. The leasing managers and front desk staffs can arrange room accordingly after the pairing deadline.

1.4 References

Centennial College COMM225 Lecture Note (2022). - Assignment #1 (Part A) Project Scope and eliciting & specifying requirements. https://e.centennialcollege.ca/d2l/le/content/795675/viewContent/9737599/View.

Fowler, M. (1997). Chapter 2. In Analysis Patterns Reusable Object Models. essay, Addison-Wesley.

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

This is a new application providing matching services for students (see fig.1), and it connects the external existing website and other relevant applications.

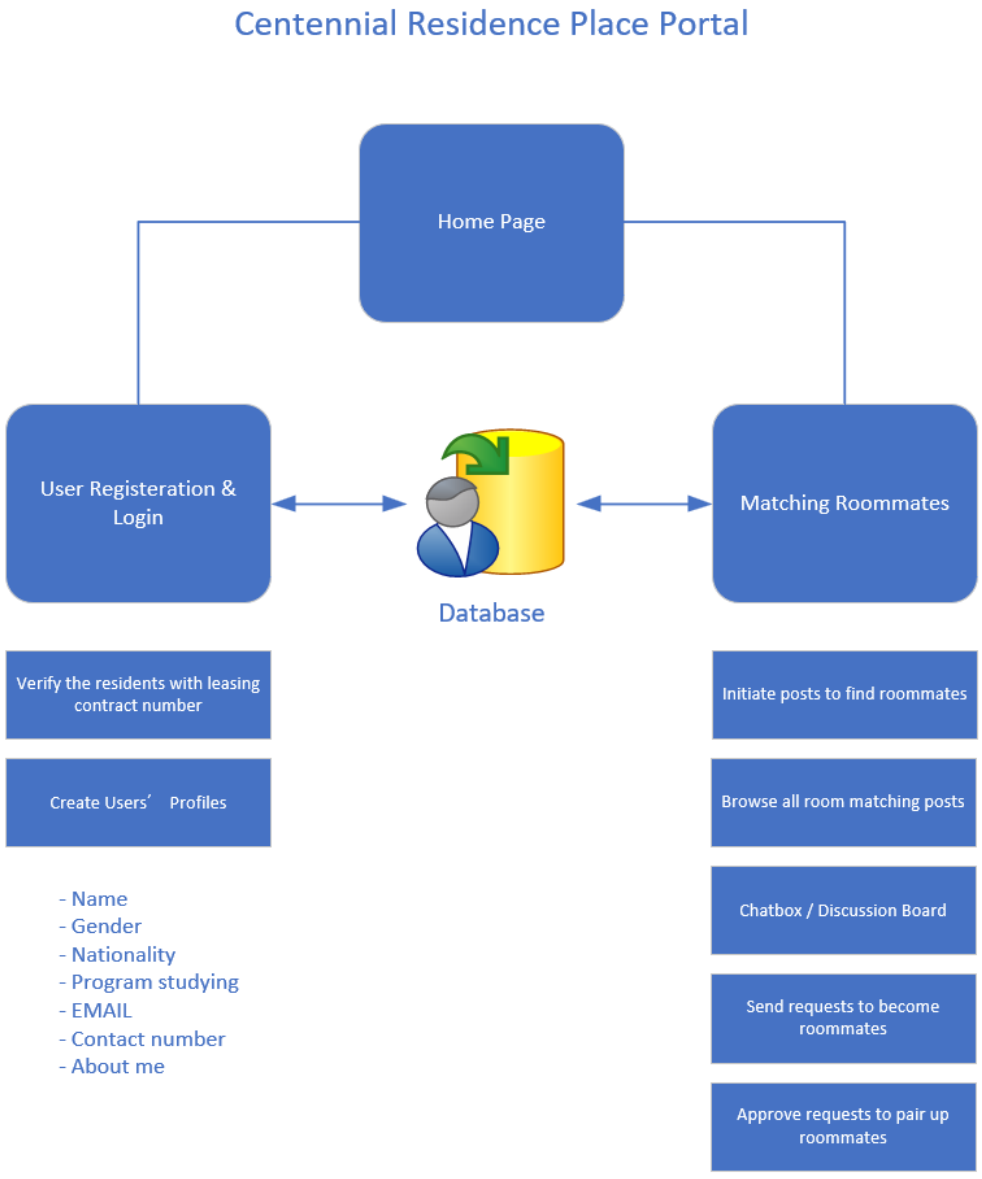


Fig 1. Major components of the Portal

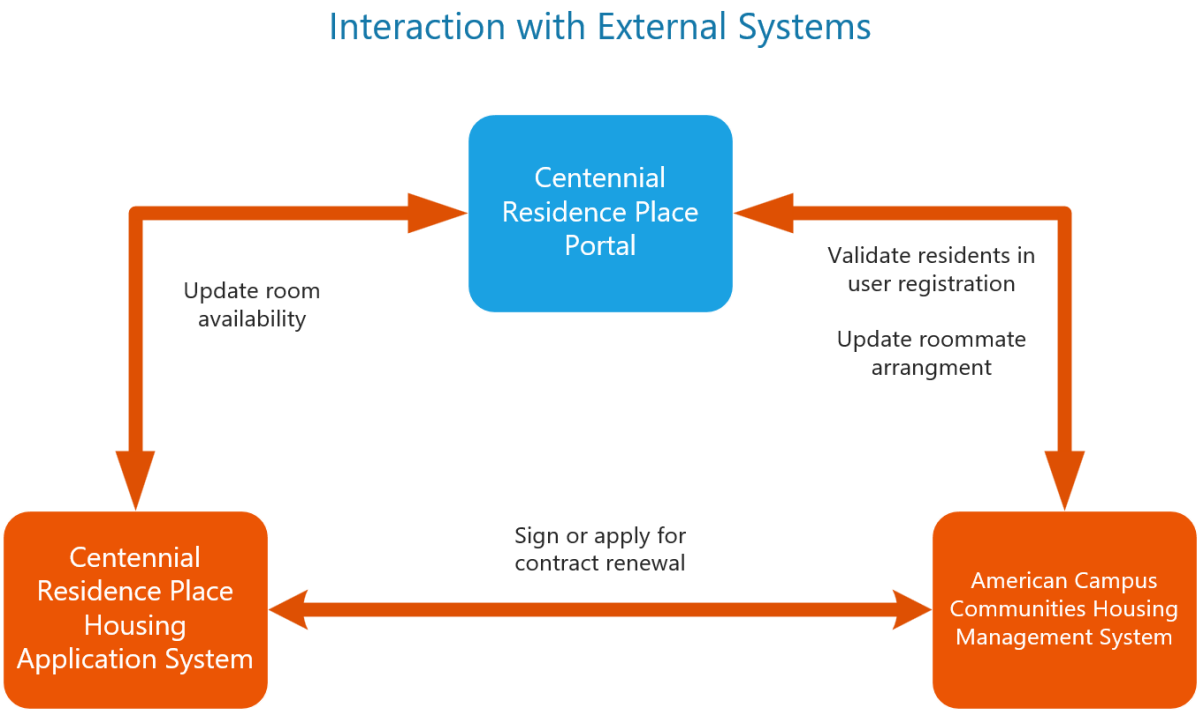


Fig 2. The interaction between the Portal and external systems

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

The software will provide the following functionalities:

- Register the student account
- Update the student profile
- Navigate other students' profiles
- Browse or create post for roommate seeking
- Communicate between other students for room pairing
- Send the result of room assignment to the students

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

The user class:
1- Students: Will use the portal to register/ navigate students' account profile, searching/ chatting with potential roommate and receive the result of room assignments
2- Lease Manager: Will use the portal to arrange the room assignments.

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

The software:

- Will run on a JBoss server
- Will use the IBM DB2 database to store the student information
- Will be developed by using JAVA
- Supports different type of the browser: Chrome, Firefox, Safari
- Support Mac, Window for personal computers
- Supports iOS, Android for mobile devices

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

Depending on the model used or the country of residence, the function of paying by card may not be supported. Currently, the dormitory receives a contract by e-mail, signs it, and then pays it. Therefore, only users who have signed the contract in advance should be able to select a room through the app.

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

User Documentation List:

- User manuals
- FAQ
- Step by step guides
- Online support team
- privacy policy
- Make Bill Payment
- American Campus Communities Contract

2.7 Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

App will work both Android and IOS so that international students can use the app without any problems. You can download it from the Google Store or Apple Store regardless of your country of residence. In addition, to minimize the problem of delay when using the app in different countries, they are all operated on the same server.

3. External Interface Requirements

3.1 User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

The user interface of this software provides students and staff with an easy-to-access platform to connect to each other. By choosing the menu on the software, students can quickly find a place to apply for their residence, choose roommates, pay the fee, or contact the staff with their questions.

- Home page

1) School images in the center of the website.
2) Header: school logo, search button, and menu line.
3) Menu line including Login, Profile(after login), Room Pairing (after login) and FAQ.

- Login Section:

1) Username, password, forget password, reset password, sign-up and help with login.
2) If the user account has not been created, students can go to sign-up page for register. Personal information including name, gender, phone number, email address, nationality, the program studying and personal description. They should also input their rental contract number for validation.

- Profile Section:

1) After login, the user can access their personal information and change their information.

- Room Pairing Section:

1) After login, students can access the Room Pairing Forum to browse all the room pairing post and create their own room pairing post.

2) Creating room pairing posts, they can input their requirements, room rules and additional personal information for matching.
3) Browsing room pairing posts, they can send request to the student who initiates the post.
4) Students can reply message under the pairing main post.
5) Post initiator can approve the pairing request.

- FAQ Section:

1) It shows the frequently asked question in this page and provides answers.

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

- Supported device types: window pc, mac pc, tablet, and mobile phone.
- Hardware needed to access software: display screen, keyboard, mouse, CPU, and memory system.
- Modem is needed to connect to Wi-Fi and network.
- GPS is needed to connect to let the application locate the students.

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

- Centennial Residence Place Portal 2022
- Users include students and staff in Centennial College.
- Allowed operating systems are Apple macOS, Microsoft Windows, Google's Android OS, Linux Operating System, and Apple iOS.
- Staff in school input the data of residence, roommate selection database, price of each type of room, dates of coming event and news, answering questions from students. *Meanwhile, students upload personal information into their student account, declaring their roommate preference, paying bills, and sending emails and messages to staff.*
- The software outputs the notification for users, displays the address of residence, and returns the command result to the user.
- Data from staff and students is stored in the database and shared across the software.

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic

forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

- The required communications functions required by Centennial Residence Place Portal 2022 include e-mail, web browser, electronic forms, and so on.
- E-mail address is required to send requests and approval for roommates' selection.
- Web browser is required to use the portal to browse for the favored roommates.
- HTTP is used to allow communication between students and school staff.
- Cryptosecurity is used to secure the creation and application of measures leading to secure ciphers and codes.

3.5 Use Case Table

Use cases			
Use Case name	**List of related Requirements ID**	**Actor (s)**	**Brief Description**
Register a student account.	FR02	Student	The actor will register the account by inputting the student number, first name, last name, id number, email address and phone number. Then the actor will receive the account by email address or SMS. Then the actor able to login the portal with the registered account.
Login the Student Account	FR02	Student	The actor to enter ID and password to login the account.
Setup the Student Profile	FR03	Student	After the actor login the portal, the actor can update the profile anytime in the main page. The actor can only update first name, last name, email address and phone number. After clicking the complete button, the email will be sent to the actor to notify the change.
Create a Room Pairing Post	FR04	Student	The actor will click on the button on forum page to create a post to seek a roommate. The actor can use an input editor to write down introduction, rules or some roommates' agreements.
Reply any Pairing Posts in the Forum	FR05	Student	The actor will leave message in the forum for the post initiator. Communicate with each other in the forum post to find the best roommate.
Send a request for pairing	FR06	Student	The actor can send a request in the posts for pairing.

Accept a pairing request to finalize the pair-up	FR06	Student	The actor who created the post can receive the request and may confirm whether to pair up or not.
View the pairing result	FR07	Leasing Manager & Management System	Leasing managers and their management system can access the result for finally arrange the rooms and roommates according to the pairing result.
Accept the pairing result	FR08	Leasing Manager & Management System	Leasing managers can accept or decline the pairing result. If there is a problem about the pairing, the manager must follow-up directly via email and phone to explain in detail.
Student identity validation	FR09	Leasing Manager & Management System	Manually, leasing managers can check the student information that the student provide in registration; or the external management system get the student information for validation.
Update FAQ	FR10	Administrator	The actor can update the FAQ with the word editor in the Portal after communicating with the Leasing managers.

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 System Feature 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

Features	Level of security (1-highest/5-lowest)	Level of priority
Login and Profile System	1	2-Medium

Room Pairing System	2	1-High
FAQ	5	3-Medium

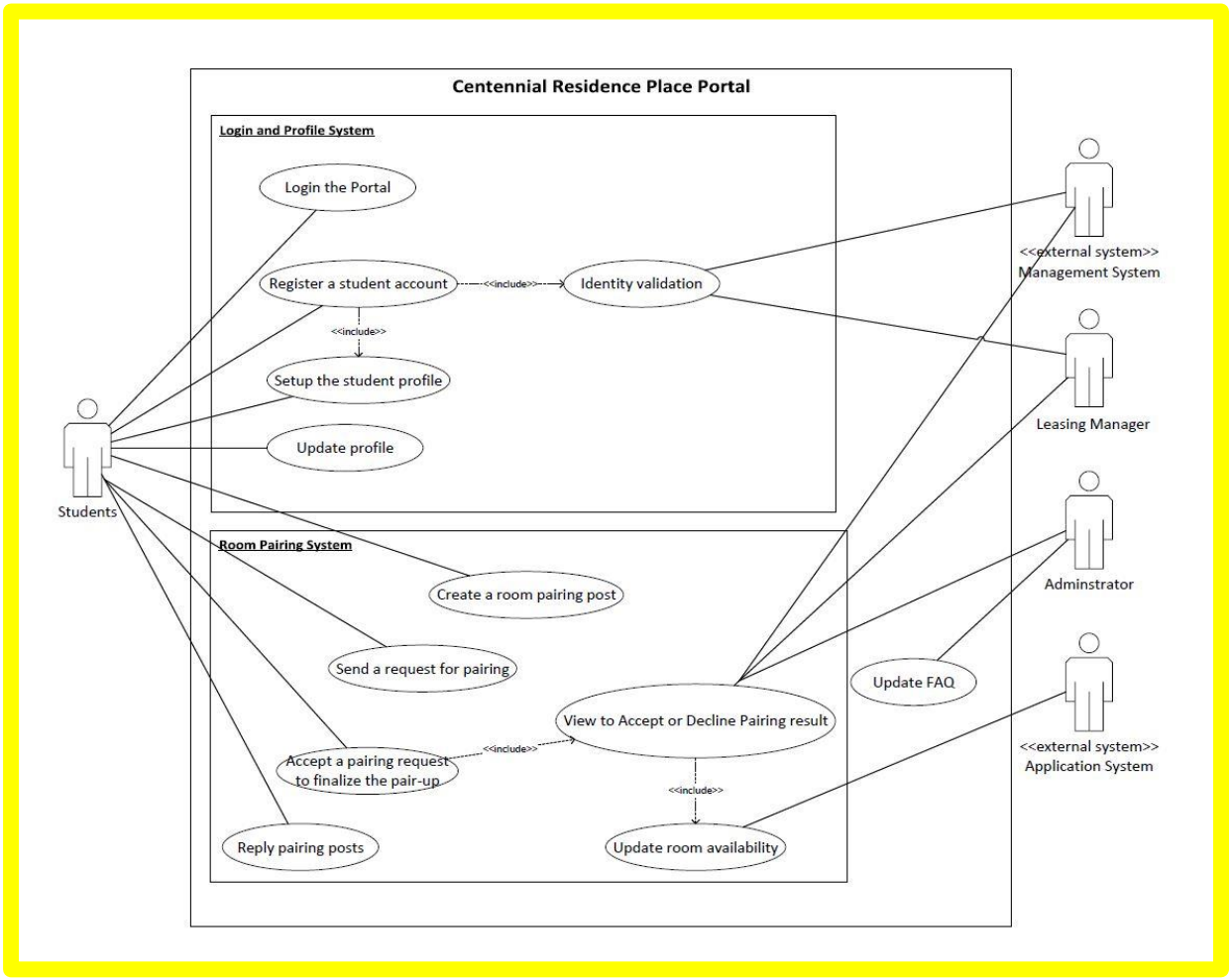


Fig 3. Use Case Diagram of the Centennial Residence Place Portal

Use case for Centennial Residence Place Portal

***Use case**: Register the student account.*
***Iteration:** 1, last modification: June 20 by group 3*
***Primary actor: student** #1*

***Goal in context:** To register in the system and enter students' personal information in order to create an account. Set up the base for logging in the portal.*

***Preconditions:** students already signed the rental contract and enable to live on campus to prepare for their next semester.*

***Trigger:** student #1 received an email from the Residence Place and it informed him to register in this app. He already downloaded centennial residence place portal*

Scenario:

1) *The student visits the Portal.*
2) *The student who has not registered should register in the register page.*
3) *The Portal displays the register page with all input blanks.*
4) *The student sets up login id and password, enters the rental contract number, and clicks on "Next".*
5) *The Portal validates the input and proceeds to the personal information page.*
6) *The student enters his/her name, gender, phone number, email address, nationality, the program studying and personal description.*
7) *The Portal shows all the setup and input to the student for confirmation.*
8) *The student clicks on "Confirm".*
9) *The Portal sent an account activation email to student's email.*
10) *The student clicks on "Activate" on the activation email.*
11) *The Portal shows "Account Created Successfully."*

***Priority:** Moderate priority, to be implemented after basic functions.*

***When available:** Available at any time except maintenance and upgrade.*

***Frequency of use:** Frequent.*

***Channel to actor:** Via window pc, mac pc, tablet, and mobile phone's browsers.*

***Secondary actors**: System administrator, GPS and cameras.*

1. *System administrator: PC-based system*
2. *GPS: is needed to connect to let the application locate the students.*
3. *Cameras: allows students to scan and upload documents.*

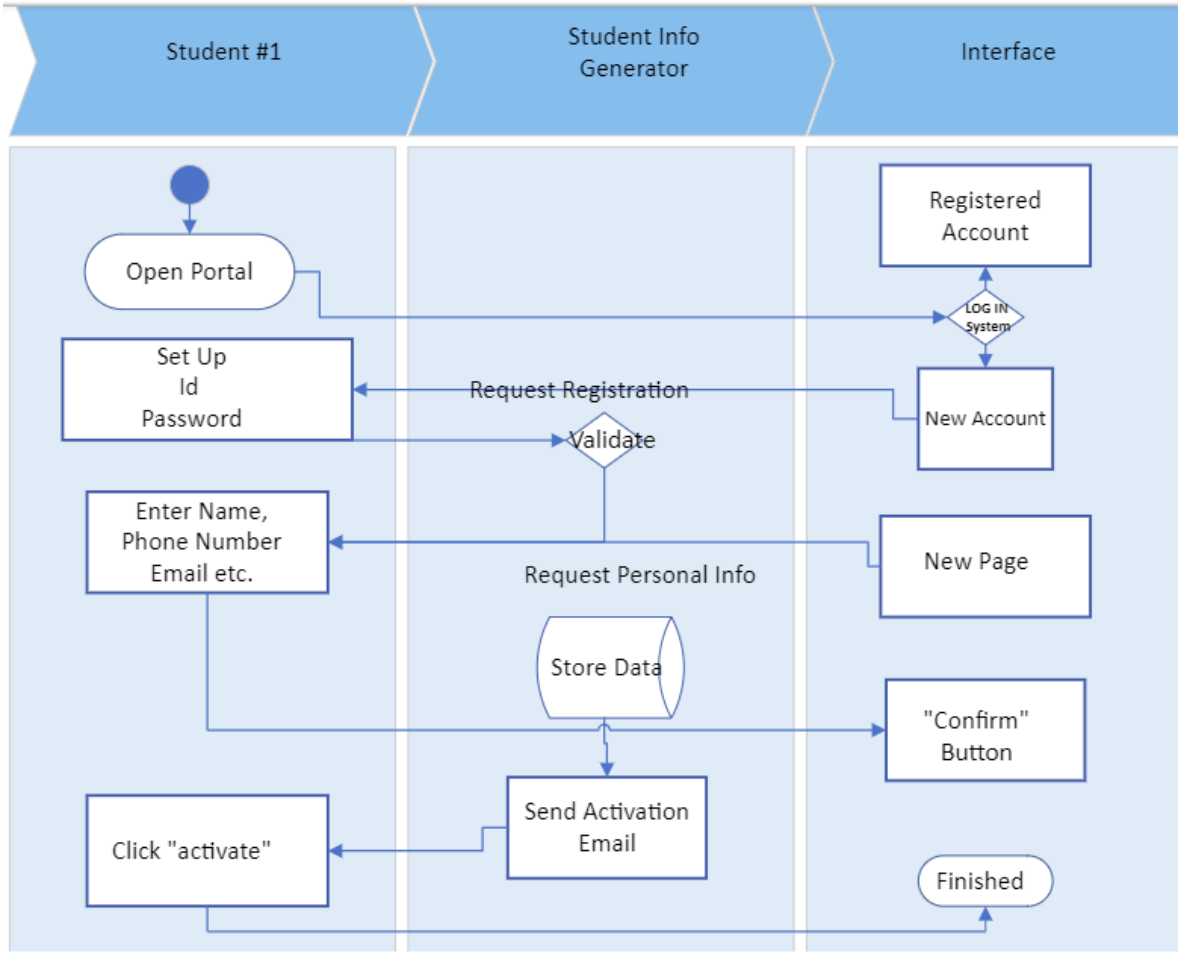
Open issues:

1. *Is the security is sufficient to ban the hacking and DDOS attack?*
2. *Is there any mechanisms protect the data collected in the system?*
3. *Will the system verify the student identity?*
4. *Will the student record will be deleted when the student is graduated or move out?*

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

SWIMLANE Diagram



4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

Functional Requirements list				
Requirement ID	**Requirement title**	**Short Description**	**Priority**	**Requester**
FR01	Alerts	The system should give the student a mobile notification/alert when: 1. a new pairing request is received; 2. pairing request is approved; 3. receiving new messages from potential roommates.	Expected	Student
FR02	Register and Login	The system should allow student to register and login.	Expected	Student
FR03	Profiles	The system should allow the students to update their profiles. Information should be shown on the room pairing post.	Expected	Student
FR04	Create Room Pairing Post	The system should allow users to create a post for room pairing, and it should allow user to write down rules and expectation for the roommates.	Expected	Student
FR05	Room Pairing Forum	The system should allow users to browse all the pairing posts in which initiators' information and expectation can be found. Users should be able to send messages to initiators.	Expected	Student
FR06	Pairing Request and Approval	The system should allow users to send a request for pairing up, and it should allow the initiators to approve the request.	Expected	Student

FR07	Pairing Result	The system should allow administrator and lease manager to view the pairing results.	Expected	Administrator Lease Manager
FR08	Frequently asked questions FAQ	The system should have the capability to allow the administrator to setup and manage frequently asked questions.	Exciting	Administrator

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:
REQ-2:

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

Non-functional requirements:

Nonfunctional Requirements list				
Requirement ID	**Requirement title**	**Short Description**	**Priority**	**Requester**
NFR01	Performance	Support 200 sessions at the same time	Expected	IT administrator
NFR02	Security	Pass the security testing: DDoS testing and malicious script testing.	Expected	IT administrator
NFR03	Mobile compatibility	Support mobiles device: iOS and Android	Expected	IT administrator
NFR04	Computer web browser compatibility	Support computer browsers: Chrome, Firefox, Safari, Edge and Internet Explorer	Expected	IT administrator

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

- Possible loss, damage, or harm including data loss and hackers. Students' information may be lost or exposed by incomplete data security protection.
- Verification code and security questions must be included to protect account security. The stealing of passwords should also be prevented.
- External policies such as Website Content Disclaimers, Privacy Policies, Intellectual Property Declaration, and so on should be included to protect the product's design and use.

5.3 Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

- Personal information of students and roommates' preference should be protected as private data.
- The information can only be shown when the roommate requirement has been approved by the data owner.
- For user identity authentication, student ID, date of birth and verification code are needed.
- Security questions must be settled when a student creates an account.

5.4 Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5 Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

Leasing Manager:
- Arrange rooms for students
- Adjust room availability

Student:
- Access to app’s certain function
- Manage their own information

Programmer:
- Back-front setting
- Give access to students, administrators and leasing managers

Database manager:
- Assign and store students' info to the system

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Abbreviation:	Explanation
RA	Room arrangement
BPS	Bill payment system
SPI	Students' Personal information storage
LRS	Log in and register system
SAS	Students' application system

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>

Below is a table to show the stakeholder register:

Stakeholder Register					
Stakeholder Name	Stakeholder Position	External/Internal	Stakeholder contact details	Operational/ Executive	Interest (high, medium, low)
John Smith	Student	External	john.smith@gmail.com	Operational	high
May Chu	Lease Manager	Internal	Maychu@com.ca	Operational	high
Nath Bob	Project manager	Internal	nath.bob@comp.ca	Operational	medium
Mary Ying	Programmer	Internal	mary.ying@comp.ca	Executive	medium

Below is a table to show the interview questions:

Interview Questions		
Question	Stakeholder position	Answer
1) What time do you normally access the portal?	Student	Before the start of the semester
2)How many times will the users to open and check it?	Leasing manager	Approximately, once a month They will check the status of the payment.
3)How does the app interact with students?	Developer	Both website and smartphone
4)How about security issues? Does it protect student's privacy?	Student	Every student will have individual files to store their information, only the developer respond to the database will have access to it.
5) How does this app accommodate the leasing schedule?	Lease Manager	Managers can set different deadlines and important dates in the app.
6) How does this app help management for room arrangement?	Lease Manager	Students will be matched before the deadline of every semester. Lease Manager may get the pairing result and consolidate the room arrangement.
7) How is the room matching run?	Student	The room matching will be run automatically based on the student preference or request by the end of submission deadline.
8) What is required before I use this portal?	Student	Students should sign the contract with the American Campus Communities and register on our portal with the contract reference number.
9) How can I find my roommate who are in the same program as mine?	Student	Students are required to enter their information while registration. They should enter their name, gender, age, nationality, program, etc. This information will be shown in the room pairing process.

10) How can I start to find my roommate?	Student	Students can start find their roommates in two ways: 1. Create pairing post, introduce yourself, enter your requirement and provide any rules for sharing room, then wait for response from your potential roommate. 2. Browse the pairing posts created by others, send request for pairing roommate.
11) How do I know if the student is a right roommate for me?	Student	Students can create rules for the share room while creating post. Students can always discuss and communicate in the chat box before sending pairing request and approving the request to confirm the pairing.
12) What if I can't pair any roommate before the deadline?	Student	The management will manually arrange a roommate for you.
13) When is the deadline of the roommate pairing?	Student	The day before the first day of the contract. Usually as same as the starting day of a new semester.
14) What programming language will be used in this portal?	Developer	Java, HTML and CSS
15) What if a student wants an empty room?	Lease Manager	Students will be able to select only non-full rooms as options unless all rooms are occupied
16) What if I want to change the room after the final selection?	Student	Once students have chosen a room, they cannot change rooms without a valid reason. A warning message will be displayed to tell this information before students make the final selection of room.
17) Can an existing student choose a new room when extending the contract?	Student	Room selection is for new students only. If you want to change rooms when extending the contract, please contact the dormitory separately.
18) Can I view my American Campus Communities contract in the app?	Student	Students can view their own contracts at any time through the Contracts menu in the app.
19) How often is the maintenance for the app?	Developer	Once a month.

20) How can I handle the enquires from the student about the app?	Administrator	Administrator can communicate with developer, project manager and lease manager accordingly to handle the enquires.
21) How should I know the room arrangement from the platform while students moving in?	Administrator	The pairing results should be sent to leasing manager for room arrange and final decision. Administrator should assist and communicate with their manager if needed.
22) How do I know if the registered students are real residents who have signed the rental contract?	Leasing Manger	Students should also provide their rental contract number for validation while registering.

Appendix E: Others

First-cut class diagram

General Classification

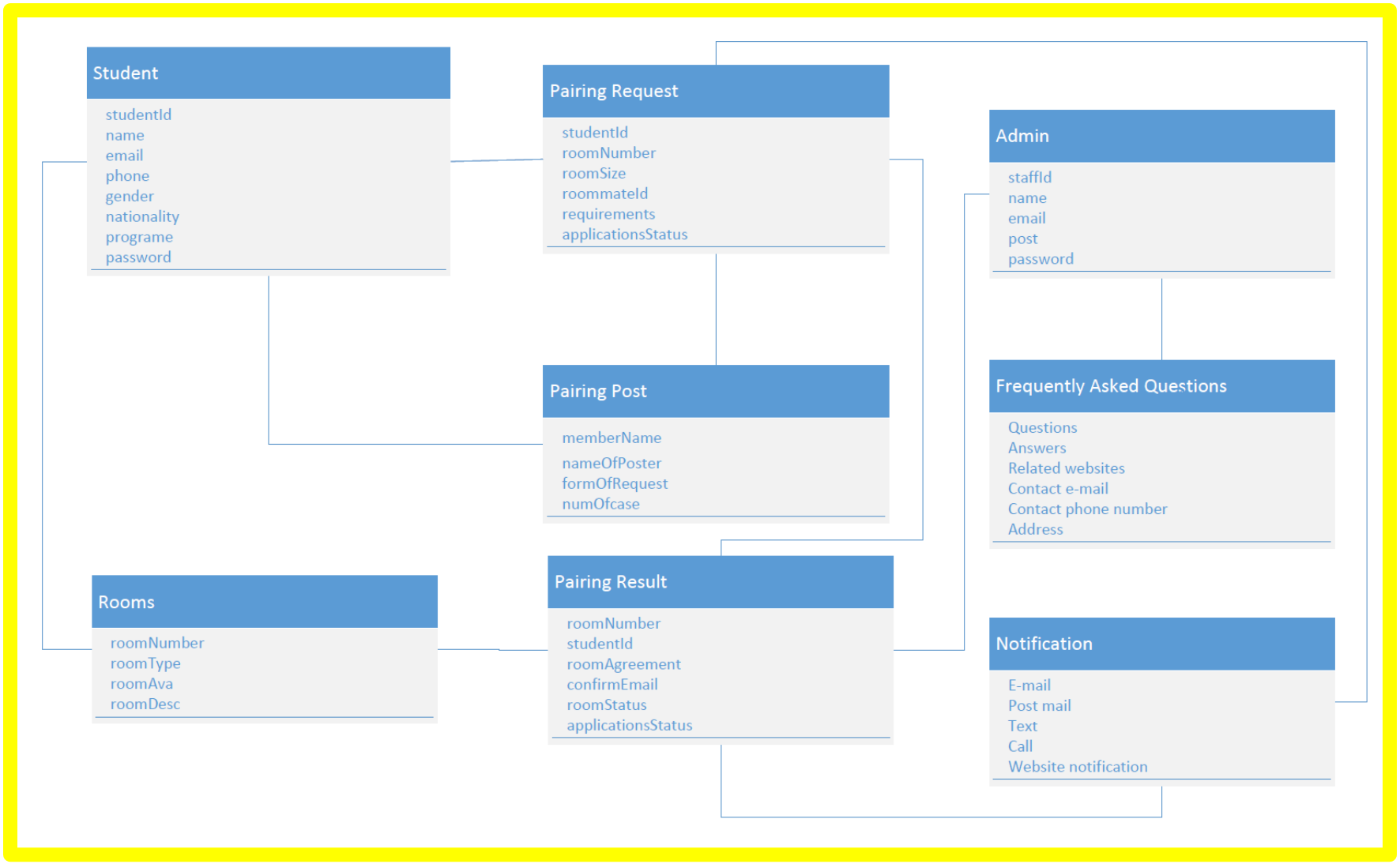
- External entities (systems, devices, people)
- Things (e.g. reports, displays, letters, signals)
- Events (occurring during system operation
- Roles (e.g. manager, engineer, salesperson)
- Organizational units (e.g. division, group, team)
- Places
- Structures (e.g. sensors, vehicles, computers)

Class Selection Criteria

1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Potential Class	General Classification	Characteristic Number That Applies
Student	role	Accepted: all apply
Roommate	role	Rejected: duplicated attributes and operations with student
Administrator	role	Accepted: all apply
Lease Manager	role	Rejected: duplicated with duplicated attributes and operations with administrator
Room	external entity	Accepted: all apply
Registration	occurrence	Rejected
Discussion	occurrence	Rejected
Student's name, phone number, gender, nationality, email and program	thing	Rejected: 3 fails, attributes for student
Pairing Post	thing	Accepted: all apply
Pairing Request	occurrence	Accepted: all apply
Pairing Result	thing	Accepted: all apply
Frequently Asked Questions	thing	Accepted: all apply

First-cut class diagram



Deliverable #3

G2_821609-538511 - Project_Group_2 - Evan Plaxton - Jun 23, 2022 1038 PM - srs_template-ieee Group 2 - Part A Resubmission

G2_821609-538512 - Project_Group_2 - Evan Plaxton - Jul 17, 2022 816 PM - srs_template-ieee Group 2 - Part B Submission-1

Attribute	Metric
1. Ambiguity	Number of ambiguous modifiers: 8 Expressions such as “various”, “many”, “expensive” and “variety” are often used. For example, in product functions, “import lyrics or music documents in various formats”, and the formats is not specified. Another example is the first use case that “a dropdown list that consists of various help needs” is not specified with the help support items. Besides, it said they are going to make a video introducing all the features in the app, but it is hard to know if it would only explain the music editing function, or if it would introduce all existing functions in the app such as recording, music copyright purchase, and listening to songs. This lack of information can give confusion when making a video.
2. Completeness	Number of TBA, TBD: 3 That is no further explanation on Wi-Fi free and pop-up free version of the app after product scope, so that is regards as TBA. Besides, most of the numbers are already well planned, for example, the app price is set up at $15. However, the maximum storage capacity of the video or maximum recording time has not been decided yet. Since most recording or editing apps limit the file size that can be processed based on the paid version, it is necessary to set this kind of number.

3. Understandability	Number of sections: 5 Number of subsections: 26 The sections/subsections are provided by the template give reader a basic understanding of the application. ~~, such as song recording, song editing, and plagiarism check, are well separated, so it is clear and predictable how the app's functions will interact with users.~~
4. Volatility	It is not an app that simply records music, but there are many functions that will be greatly affected by changes in systems and standards, such as purchasing music authorship and detecting plagiarism. These features have high volatility which could lead to removing some functions. Furthermore, a function such as a plagiarism detection app does not yet have an exact standard for plagiarism itself, and it cannot be called plagiarism simply because the melody is similar, so it can create a lot of controversies. Expecting the high volatility of developing this app, but the "Revision History" in content page is missing. Thus, the number of changes for this project is unknown.

5. Traceability	Number of requirements not traceable to design/code: 5 1. Users can purchase rights to use music content, the use case for purchasing in use case diagram is missing and untraceable. 2. The follow functions mentioned in the diagram in 2.1 Product Perspective are missing and cannot be found in the functional requirements list, use case diagram and class diagram: 2.1) Play music 2.2) Share their music in Instagram 3. The “Recommendation” function is suddenly introduced in use case list. It is not included in 2.1 Product Perspective and is not followed up in class diagram 4. The “Login” function is not traceable without a sign up/ registration function. ~~It is possible to check the figures such as which songs the user purchases and how much they use, but due to personal privacy issues it is difficult to accurately check how well the user uses the song recording and editing functions, which are the main functions of the app.~~

6. Model clarity	Number of UML models: 3 Number of descriptive pages per model: 1 Number of UML errors: 4 The swim lane diagram has no title or caption to identify the diagram. In use case diagram, “recommendation” is not an action. It is be specified as “view recommendation” and “recommend music content”. In class diagram, “Purchase” class should not be the child class of “Musician”. In “Musician” class, function “+createTrack()” returns a string as “void” is contradicted. ~~Interrelationships such as the relationship between the app and the server and between the server and the user were clearly displayed. Model clarity is high as it is clearly expressed in a diagram that shows how users can use the app and how they can use the function.~~

CRC Index Cards

Class: Admin	
Responsibility:	**Collaborator:**
Validate the student account application	Student
View the pairing result	Pairing result
Update Frequent Ask Questions	Frequent Ask Questions
Create admin account	
Account login	
Account logout	
Update account info: email and password	
Forget password	

Class: Student	
Responsibility:	**Collaborator:**
Login	
Logout	
Update account info: email, password, phone and program	
Register a student account	
Setup the Student Profile	
Delete Student Account	Pairing Post
Forget Password	Pairing Request

Class: Room	
Responsibility:	**Collaborator:**
Dividing rooms into different types	
Generating number of rooms and naming them a specific number	
Update room avaliability based on room pairing Post	Pairing result
Add room description	
Restrict the creation of pairing request according the room availabilty	Pairing Request

Class: Pairing Post	
Responsibility:	**Collaborator:**
Create pairing post	Student
Edit pairing post	Student
Reply and feedback the post	Student
Check the room avaliabilty	Room
Combine the avaliable rooms and students request	Pairing Request

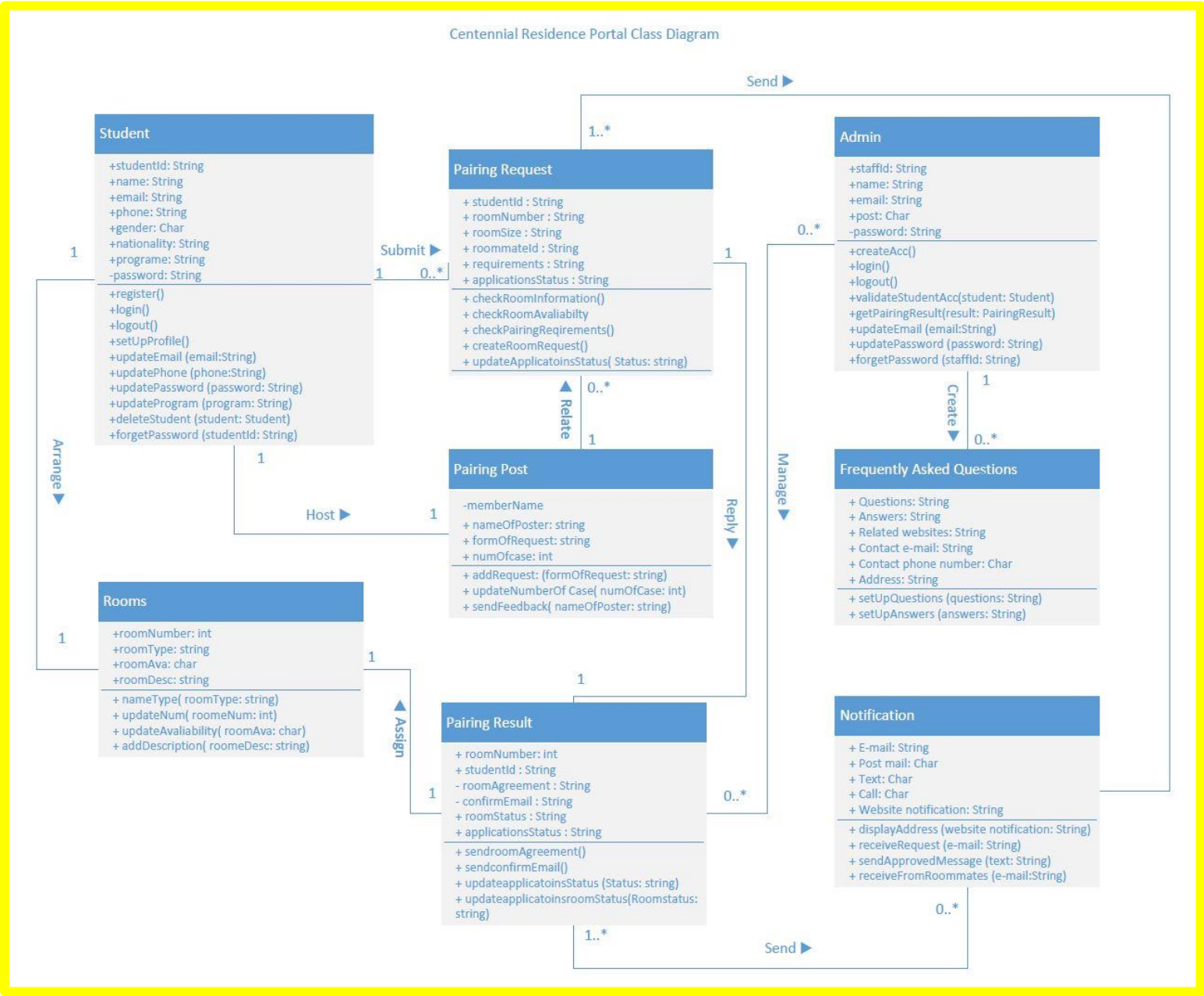
Class: Pairing Request	
Responsibility:	**Collaborator:**
Check room information	Room
Create and send pairing request	Student
Check the room avaliabilty	Room
Proceed with the request	

Class: Pairing Result	
Responsibility:	**Collaborator:**
Send confirm Email to student	Student
Send room agreement to student	Student
Change student status to contract complete	Student
Change room status to selected	Room

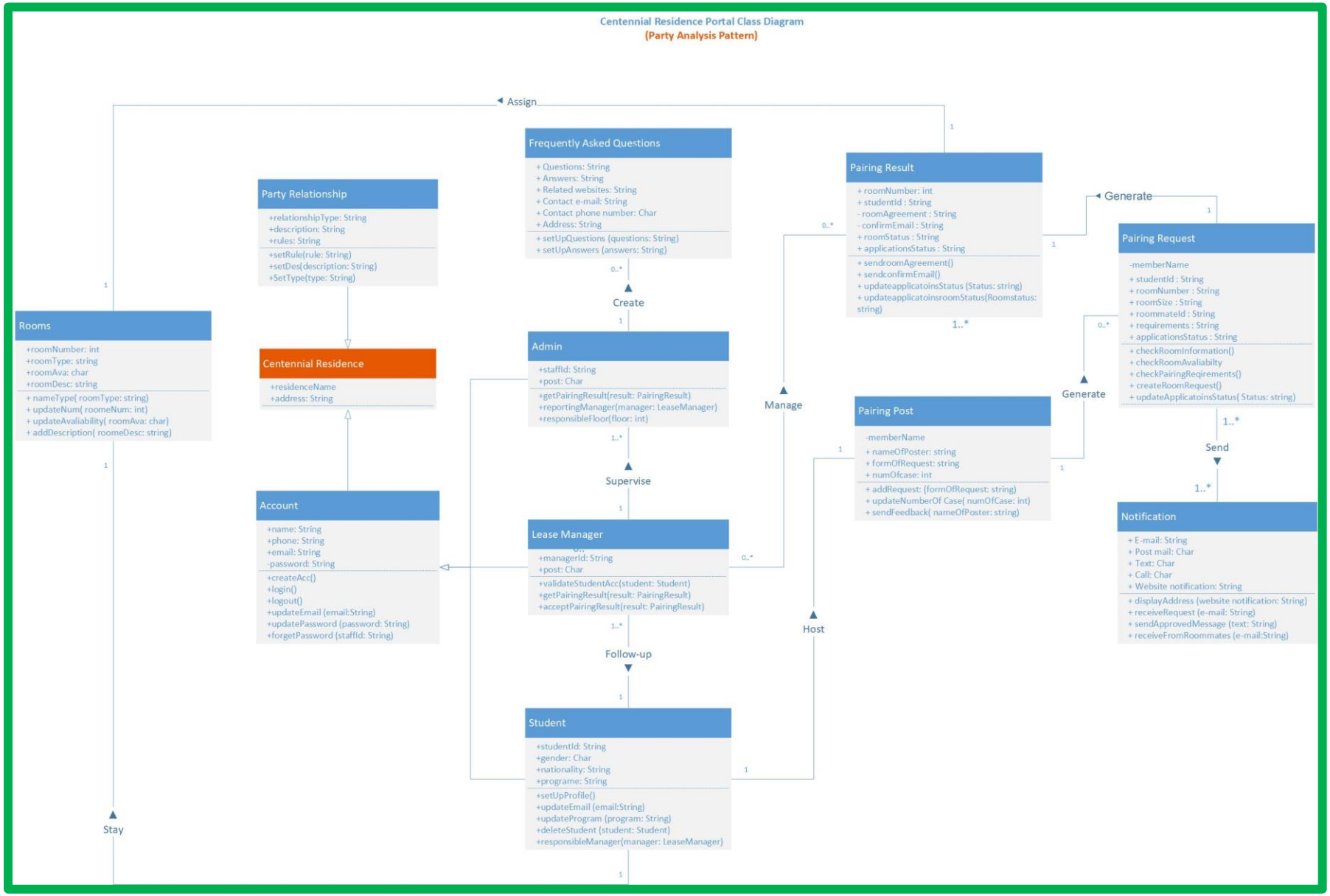
Class: Notification	
Responsibility:	**Collaborator:**
Displaying the address of the residence	
Returning the command result to the user	
Receiving a new pairing request	Pairing Request
Sending an approved message from pairing request	Pairing Request
Receiving new messages from potential roommates	

Class: Frequently Asked Questions	
Responsibility:	**Collaborator:**
Allow to setup frequently asked questions	Admin
Manage frequently asked questions	Admin
Update frequently asked questions	Admin
Communicate with leasing managers	Admin

Domain class diagram

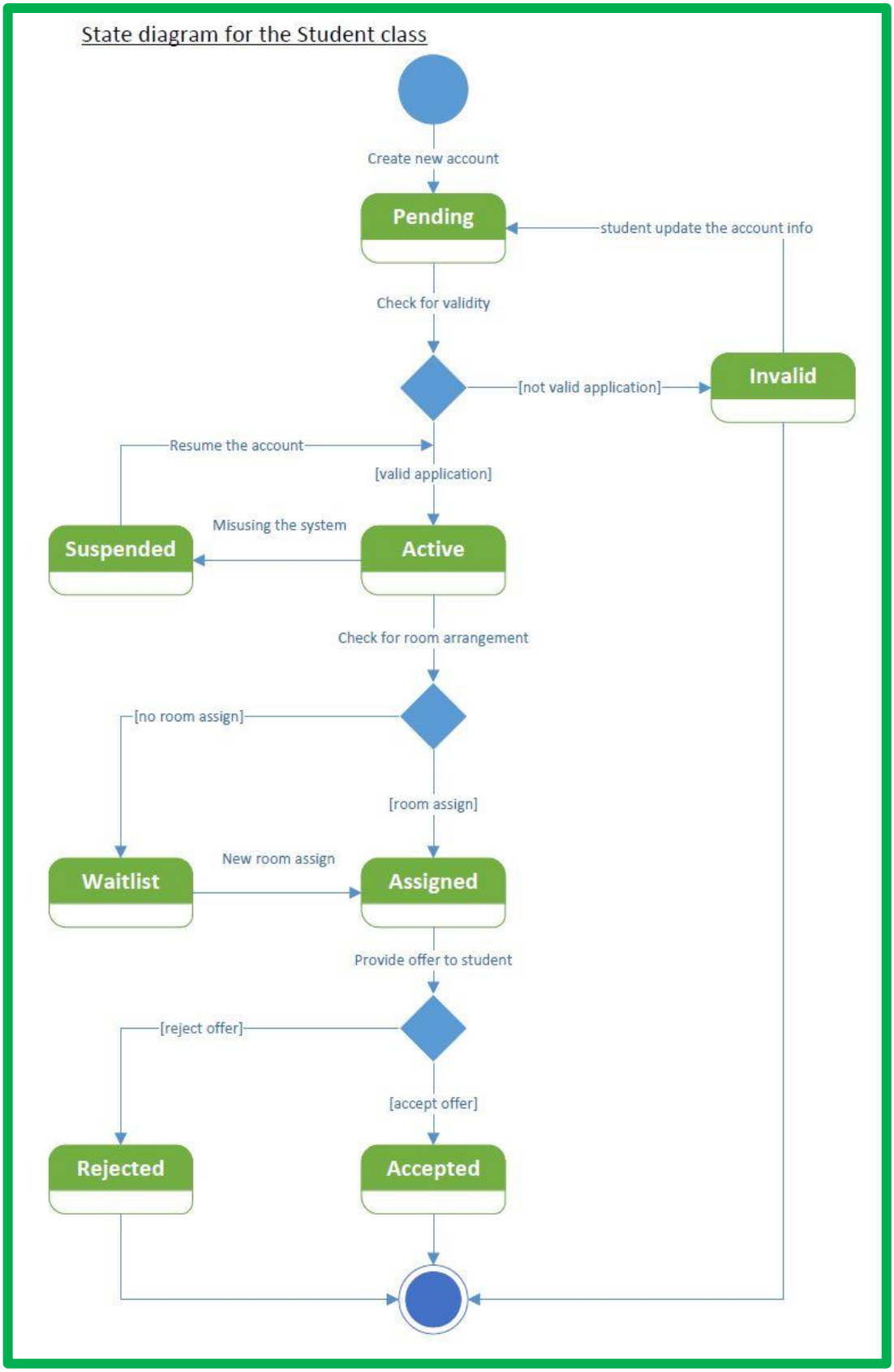


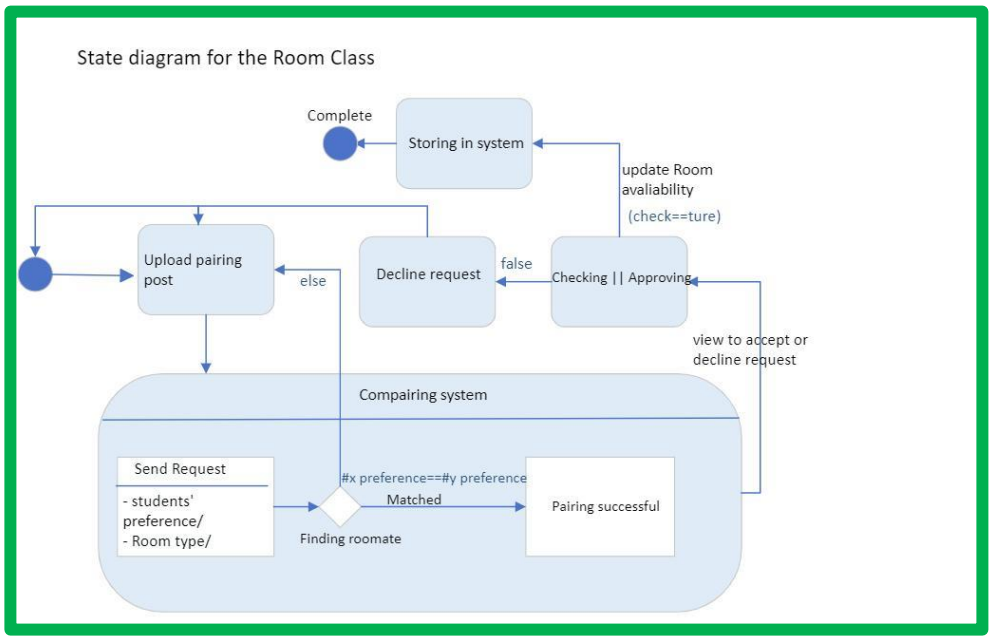
Domain class diagram (Party Analysis Pattern)



According to the party analysis pattern by Martin Fowler(1996), this pattern is useful for organization that emphasizes the concept of accountability. While Centennial Residential is managed by the management agency and responsible to their residents (the students), the model can be used for the management structure in the Portal. Centennial Resentence will be the “Party”, in which “Party Relationship” covers the accountability for the lease manager, administrator and students. As the original class diagram has a vague relationship between the lease manager and administrator, there are no distinguished classes of lease manager and administrator. Revising the diagram with the party analysis pattern, an accountability is modeled that a lease manager is type of manager, with administrators report to the lease manager. At the same time, an accountability is modeled that a lease manager follows up and response to students accordingly.

State Diagram




State diagram for the Room Class
Complete
Storing in system
update Room
avaliability
(check==ture)
Upload pairing
post
else
Decline request
false
Checking || Approving
view to accept or
decline request
Compairing system
Send Request
- students'
preference/
- Room type/
#x preference==#y preference
Matched
Finding roomate
Pairing successful

Sequence Diagram