Seat Planning System

Software Requirements Specification and Analysis

BUS-602: Management Information Systems

Submitted to

Dr. Md . Mahbubul Alam Joarder

Professor

Institute of Information Technology

University of Dhaka

Amit Seal Ami

Lecturer

Institute of Information Technology

University of Dhaka

Submitted by

- Nasimuzzaman Himel
- Ali Zafar Sadiq
- A. S. M. Khairul Alam
- Naushad Hossain
- Sazidul Islam

Submission Date

17/11/2016

LETTER OF TRANSMITTAL

November, 2016

Dr. Md . Mahbubul Alam Joarder

Professor

Amit Seal Ami

Lecturer

IIT, University of Dhaka

Sir,

We have prepared the enclosed report on Software Requirements Specification of **'Seat Planning System'** for your approval. This report details the requirements I gathered for the project.

The primary purpose of this report is to summarize our findings from the work that we completed as our Software Requirements Specification and Analysis course project. This report includes the details of each step we followed to collect the requirements.

Sincerely Yours,

BSSE0602-Nasimuzzaman Himel

BSSE0603-Ali Zafar Sadiq

BSSE0612-A. S. M. Khairul Alam

BSSE0625-Naushad Hossain

BSSE0633- Sazidul Islam

3rd Year, 6th Semester, 6th Batch

Session: 2013-2014

Institute of Information Technology

University of Dhaka

Executive Summary

The purpose of Seat Planning System, which is a web application, is to help users (mainly teachers) to arrange seat plan for exams. It takes student rolls and randomizes them. It also takes teachers assigned to the room and informs the teacher through mail. Besides here if the user gives different length of columns wise seats number, then rolls will also be arranged likely.

Acknowledgement

We express our heartiest gratitude to Almighty ALLAH to complete our designated SRS report in time and without hassles. We are grateful to our honorable sir Dr. Md. Mahbubul Alam Joarder and Amit Seal Ami sir for their help throughout the working time. They helped us a lot by sharing their knowledge and guiding us to the right direction.

Contents

Chapter 1: Introduction

This chapter is intended to specify the purpose of this document and the intended audiences of it.

1.1 Purpose

This document is the Software Requirement Specification (SRS) for the Seat Planning System. It contains functional, non-functional, support requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely identified, numbered, and organized by topic. The SRS serves as official means of communicating user requirements to the developer and provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

This SRS is intended for several audiences including the customers as well as the project managers, designers, developers, and testers.

- The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
- The project managers of the developer team will use this SRS to plan milestones a delivery date, and ensure that the developing teams on track during development of the system.
- The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developers will use this SRS as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created software that will fulfill all of the customer's documented requirements.

- The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure.

Chapter 2: Inception

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project. To establish the groundwork we have worked with the following factors related to the inception phases:

- Identifying Stakeholders
- Asking the First Questions
- Recognizing multiple viewpoints
- Working towards collaboration

2.1.1 Identifying Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. To identify the stakeholders we consulted with Assistant exam-controller and asked him following questions:

- Who is paying for the project?
- Who will be using the project outcomes?
- Who gets to make the decisions about the project (if this is different from the money source)?
- Who has resources I need to get the project done?
- Whose work will my project affect? (During the project and also once the project is completed)?

Concluding thoughts on Stakeholders**,** We identified following stakeholders for our automated pre-exam control system**:**

Teacher: Teacher is the person who is the only user in our system. He/She is a member of Exam Committee. He/she needs to register with e-mail address and password. He/she has the ultimate authority to create a seat plan.

2.1.2 Asking the First Question

We set our first set of context-free questions focuses on the customer and other stakeholders, overall project goals and benefits. The questions are mentioned above. These questions helped us to identify all stakeholders, measurable benefit of the successful implementation and possible alternatives to custom software development. Next set of question helped us to gain a better understanding of problem and allows the customer to voice his or her perception about the solution. The final set of question focused on the effectiveness of the communication activity itself.

2.1.3 Recognizing Multiple Viewpoints

As Teacher is our only stakeholders, so we need to consider only his/her viewpoints. We collect these viewpoints by discussing with Amit Seal Ami sir from Institute of Information Technology (IIT), University of Dhaka (DU).

His viewpoints:

- Web-based Interface.
- Unbiased random seat planning.
- Easy Access.
- Strong Authentication.
- Accessible from any computer that has internet access.
- Accessible via mobile with internet access.

2.1.4 Working towards collaboration

In order to work towards collaboration, there are some steps to follow

- Identify the common and conflicting requirements.
- Categorize the requirements.
- Take priority points for each requirements from stakeholders and on the basis of this voting prioritize the requirements.
- Make final decision about the requirements.

As in our system, our stakeholder is only a single person, we didn't have to follow these steps. We consider his viewpoints as our final requirements.

2.1.5 Conclusion

Inception phase helped us to establish basic understanding about book circulation system in a library, identify the people who will be ben**e**fited if book circulation system becomes automated, define the nature of the book circulation software and establish a preliminary communication with our stakeholders.

Chapter 3: Elicitation

3.1 Introduction

Elicitation is a task that helps the customer to define what is required. To complete the elicitation step we face many problems like problems of scope, problems of volatility and problems of understanding. However, this is not an easy task. To help overcome these problems, we have worked with the Eliciting requirements activity in an organized and systematic manner.

3.2 Eliciting Requirements

Unlike inception where Q&A (Question and Answer) approach is used, elicitation makes use of a requirements elicitation format that combines the elements of problem solving, elaboration, negotiation, and specification. It requires the cooperation of a group of end-users and developers to elicit requirements. To elicit requirements, we completed following four works.

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenarios
4. Elicitation work products

3.3 Collaborative Requirements Gathering

We completed following steps to do it.

- The meetings were conducted with Amit Seal Ami Sir, lecturer at Institute of Information Technology, University of Dhaka as well as member of exam committee. He was questioned about his requirements and expectations from the Seat Planning System.
- He was asked about the problems he is facing with the current manual system.
- At last we selected our final requirement list from the meetings.

3.4 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software .It concentrates on maximizing customer satisfaction from the Software engineering process .With respect to our project the following requirements are identified by a QFD.

3.4.1 Normal Requirements

1. Accessible via the Internet.
2. Unbiased random seat planning.
3. Strong authentication.

3.4.2 Expected Requirements

1. Downloadable as pdf format.
2. User manual.
3. Handle dummy seat.

3.4.3 Exciting requirements

1. Take input from excel.

3.5 Usage Scenarios

Seat Planning System

The Seat Planning System (SPS) is proposed by Institute of Information Technology (IIT), University of Dhaka (DU). SPS is a web based software application that will be used to automate the existing manual seat planning system.

For using this system, user needs to register with e-mail address and password. After registering user needs to verify his email address by clicking link sent to that email address. Then user can log into the system using email and password. After that he or she will be allowed to create seat plan.

Seat Planning System will take roll numbers, room number, and room's size in maximum rows and columns, students email addresses (optional), number of students to be placed in a column (optional) as input and will randomize rolls and map them to seats so that no adjacent roll students are placed nearby of a roll placed in a seat. In Case of uneven rows and columns one will have to mention that which row or column with its size. After placing seats user can modify the seat plan by exchanging rolls between seats or with dummy seat. Seat Planning System will also take teachers name who will perform duty on that room. After randomizing and allocating seats, user will be given choice whether he wants to send seat plan to students (if email addresses of students are provided).

3.6 Elicitation work product

The output of the elicitation task can vary depending on size of the system or product to be built. Our elicitation work product includes:

- Make a statement of our requirements for Seat Planning System.
- Make a bounded statement of scope for our system. Make a list of customer, user and other stakeholder who participated in requirements elicitation.
- Set of usage scenarios.
- Description of the system's technical environment

Chapter 4: Scenario Based Modeling

Scenario-based modeling depicts how the user interacts with the system and the specific sequence of activities that occur as the software is used. This chapter describes the scenario based modeling for the Seat planning System of IIT, University of Dhaka.

4.1 Definition of Use case

A use case is a series of related interactions between a user and a system that enables the user to achieve a goal. A use case diagram simply describes a story using corresponding actors, who perform important role in the story and makes the story understandable for the users.
The first step in writing a use case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using system.

Primary Actor:
Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

Secondary Actor:
Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

4.2 Use Case Diagrams

Use case diagrams give the non-technical view of overall system.

4.2.1 System Description from Level-0 use case:

After analyzing the user story we found one actors who will directly use the system as a system operator. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information. Following is the actor of SPS system–

 User

Here in our SPS system we have one actor interacting with the system:

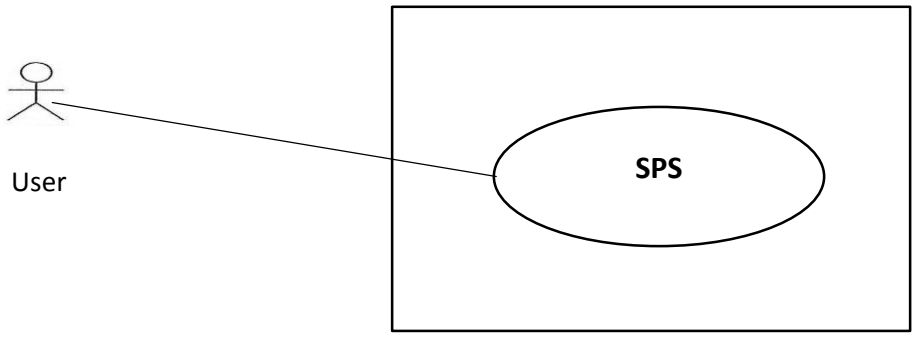


Figure: Level 0

4.2.2 System description from level-1 use case diagram:

The actors of SPS have to play different actions and system will reply according to these actions –

User:

Action1: Click signup.
Reply1: Please fill up the required information.

Action2: Enters the information.
Reply2: Registration successful.

Action3: Enters username and password.
Reply3: Login successful.

Action4: Giving required input for seat planning.
Reply4: Input taken.

Action5: Click on create seat plan.
Reply5: Seat plan created.

Action6: Interchanging values of Seat Plan.
Reply6: Seat Plan Modified.

Action7: Including teacher's duty.
Reply7: teacher's duty assigned.

Action8: Sending email to students regarding seat plan.
Reply8: Email sent.

Action9: Downloading seat plan.
Reply9: Pdf of seat plan downloaded.

Level-1 use case diagram for Seat Planning System:

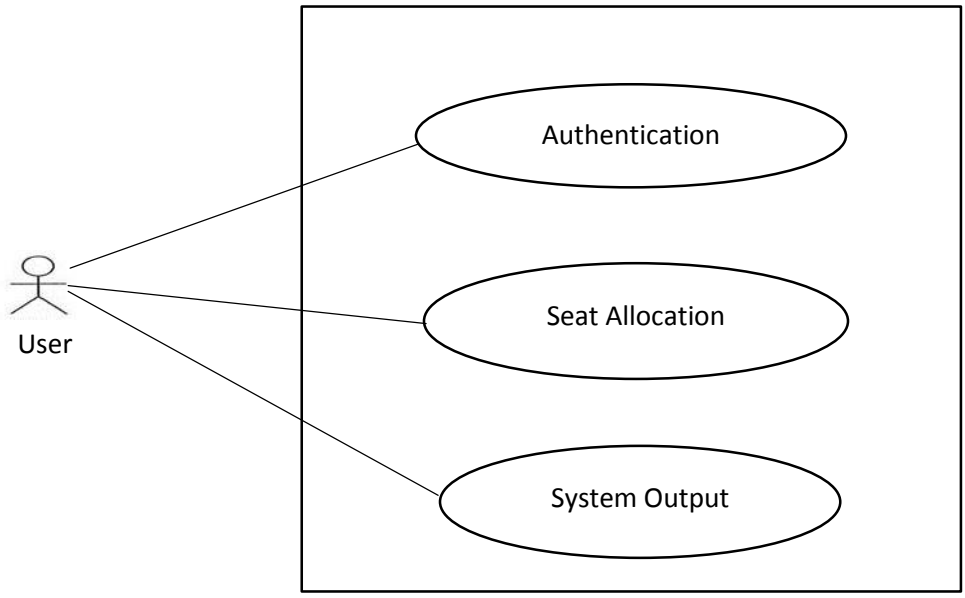


Figure: Level 1

Subsystems of Authentication: If any actor wants to perform any task then he needs to go through the authentication process. At first the actor need to have an account if they want to interact with the system. If they have not any account then they need to sign up.

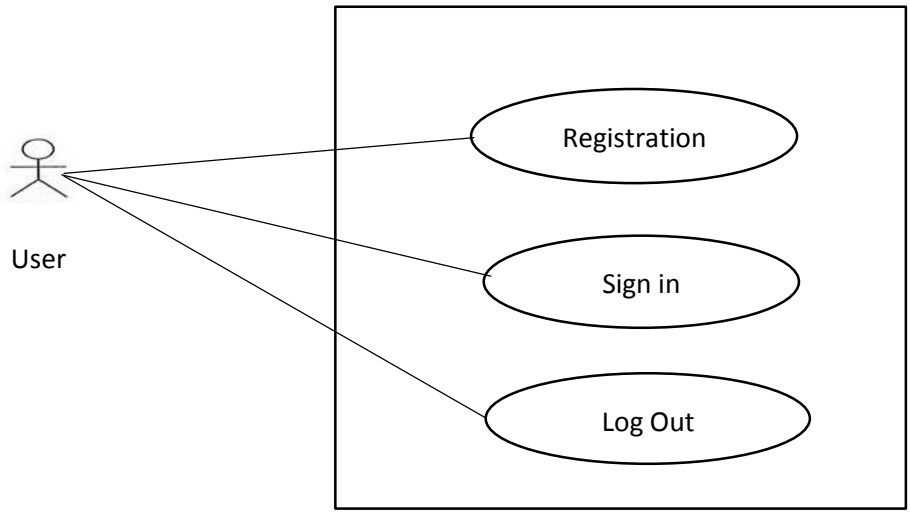


Figure: Level 2.1

Subsystems of Seat Allocation: User have to give required input such as room number, column number, students number to make a seat plan. Then User can modify the seats and including teacher's duty to the seat plan.

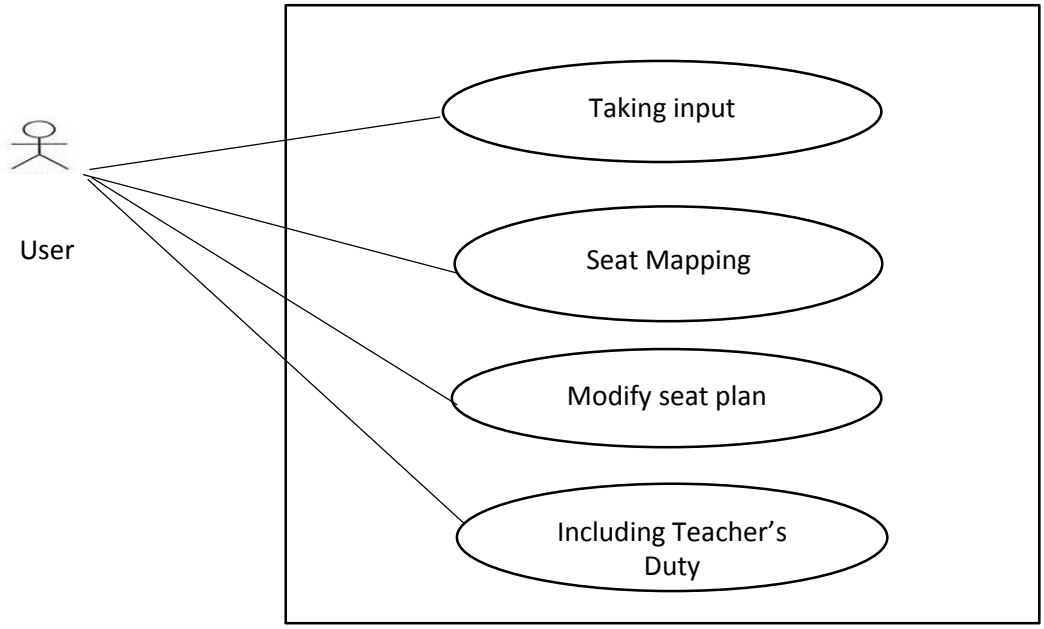


Figure: Level 2.2

Subsystems of System output: User can send mail to the students about their exam seat plan. User can also download the seat plan as pdf format.

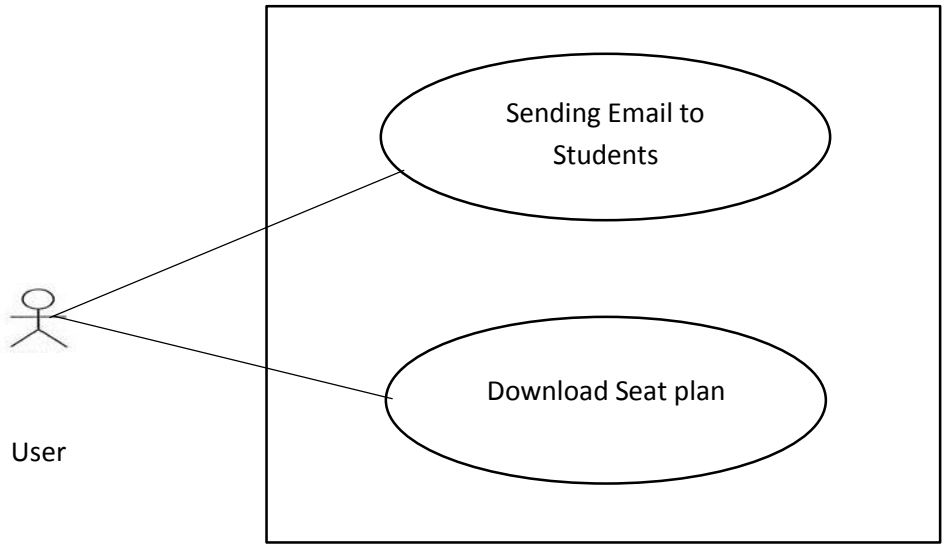


Figure: Level 2.3

4.3 Activity & Swim lane Diagrams

4.3.1 Activity Diagram

An activity diagram represents the actions and decisions that occur as some function is performed.

4.3.1.1 Activity Diagram for user registration

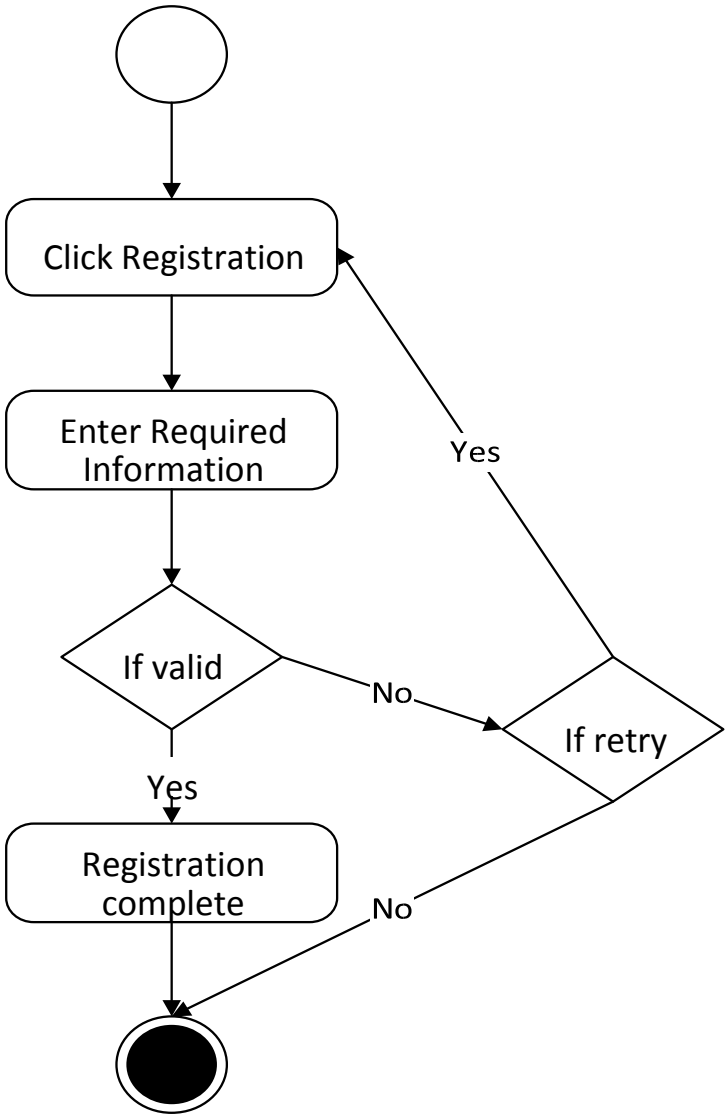


Figure: Activity diagram of user registration

4.3.1.2Activity Diagram for user Login

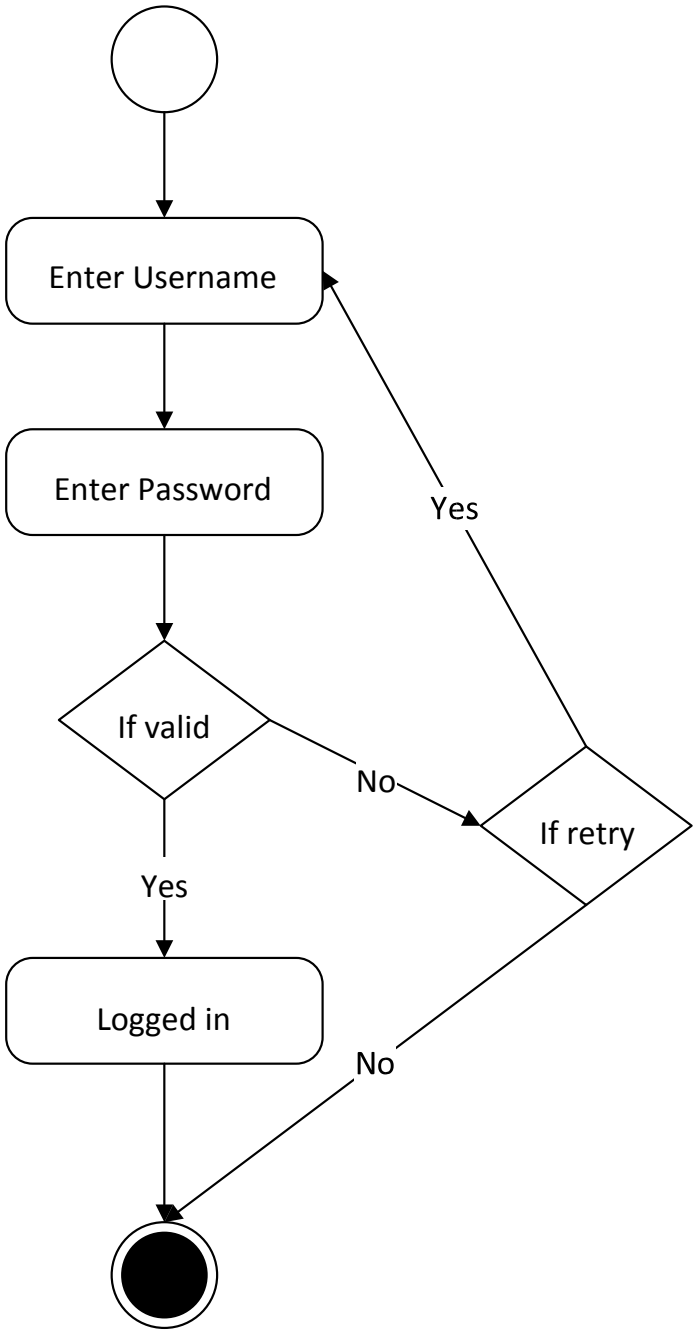


Figure: Activity diagram of user login

4.3.1.3 Activity Diagram for logged out:

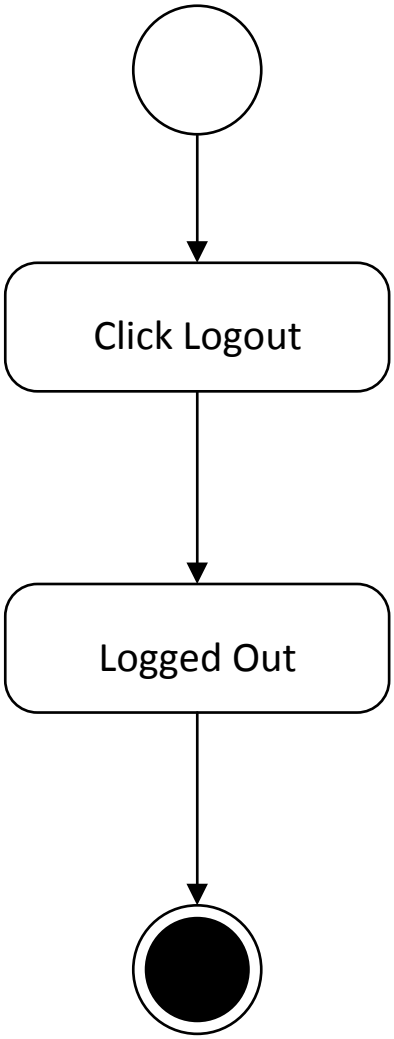


Figure: Activity diagram of user logout

4.3.2 Activity Diagram for Seat Allocation:

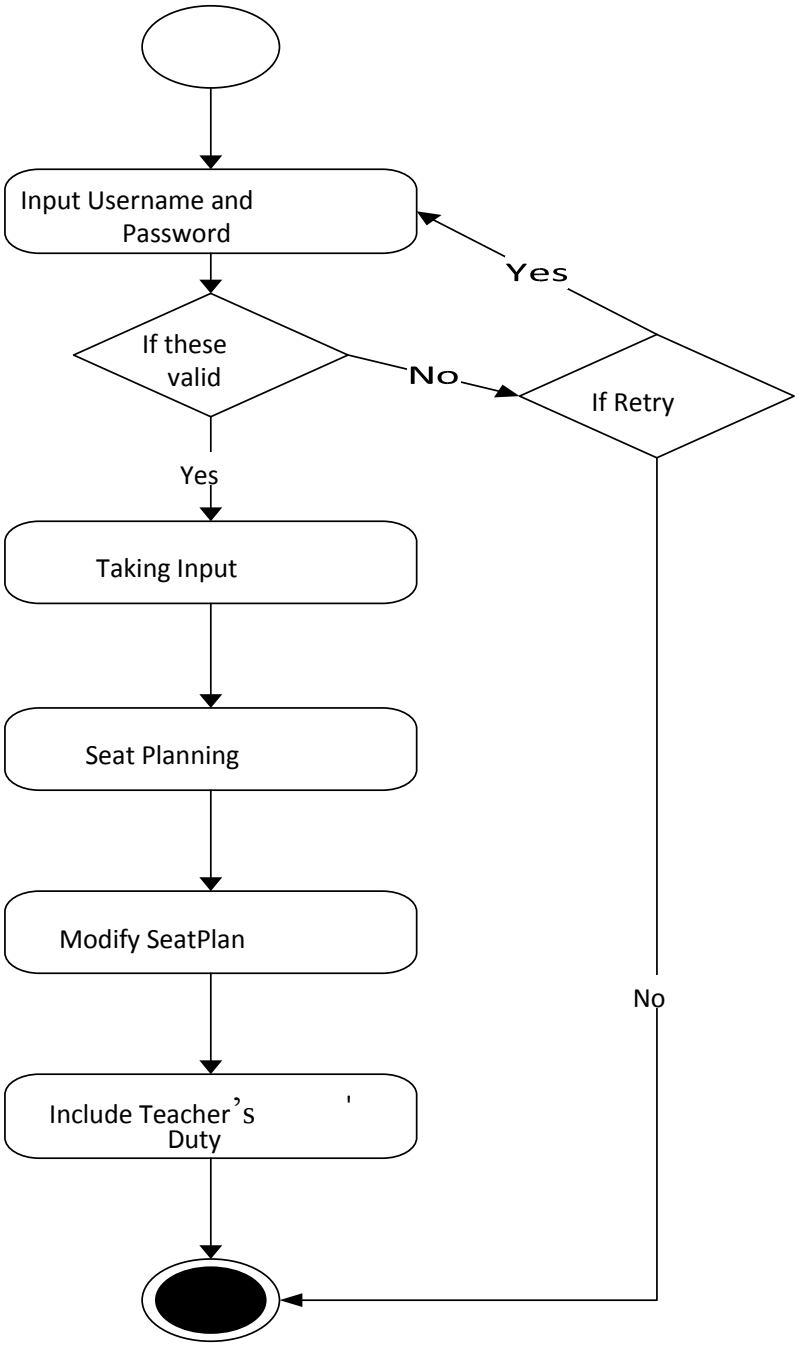


Figure: Activity Diagram for Seat Allocation

4.3.3 Activity Diagram for System Output:

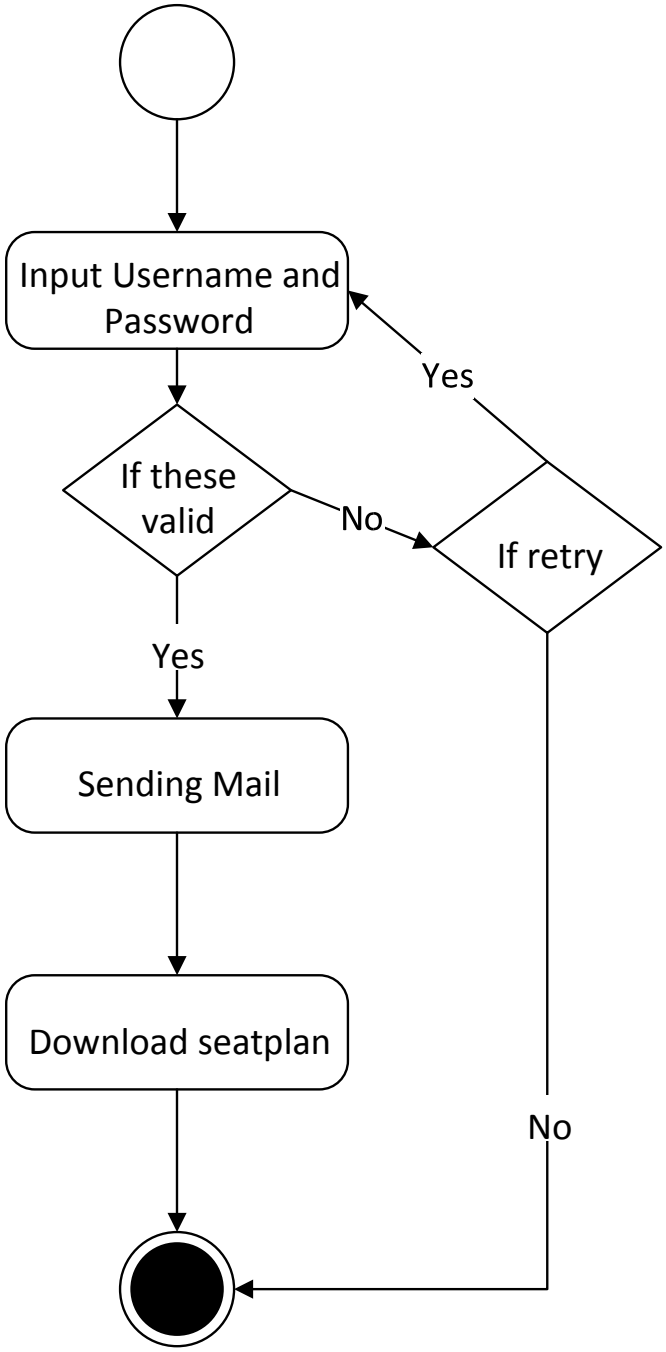


Figure: Activity Diagram for System Output

4.3.2 Swim Lane Diagram

4.3.2.1 Swim lane diagram for user registration

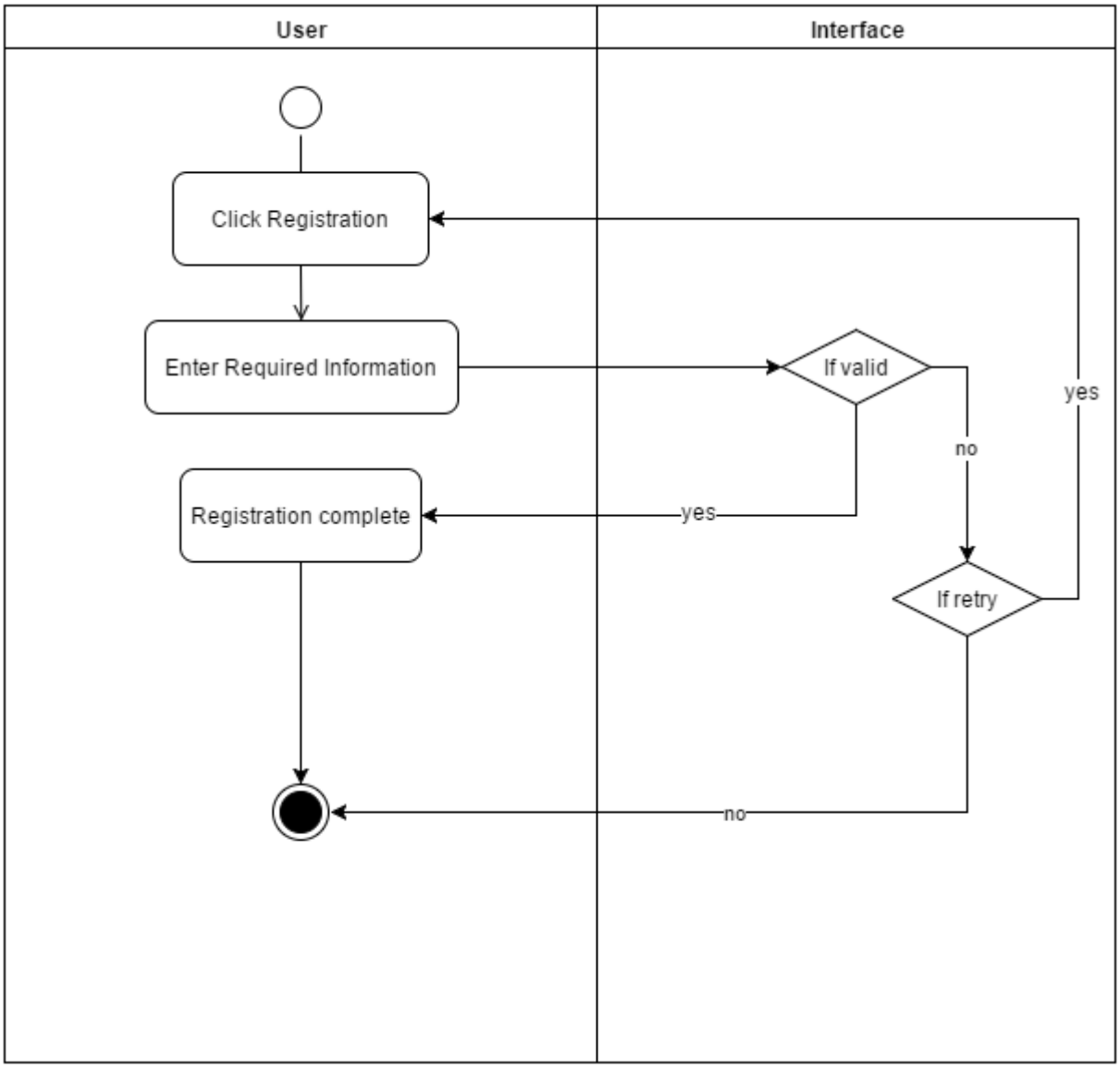


Figure: Swim lane diagram for user registration

4.3.2.1 Swim lane diagram for user login

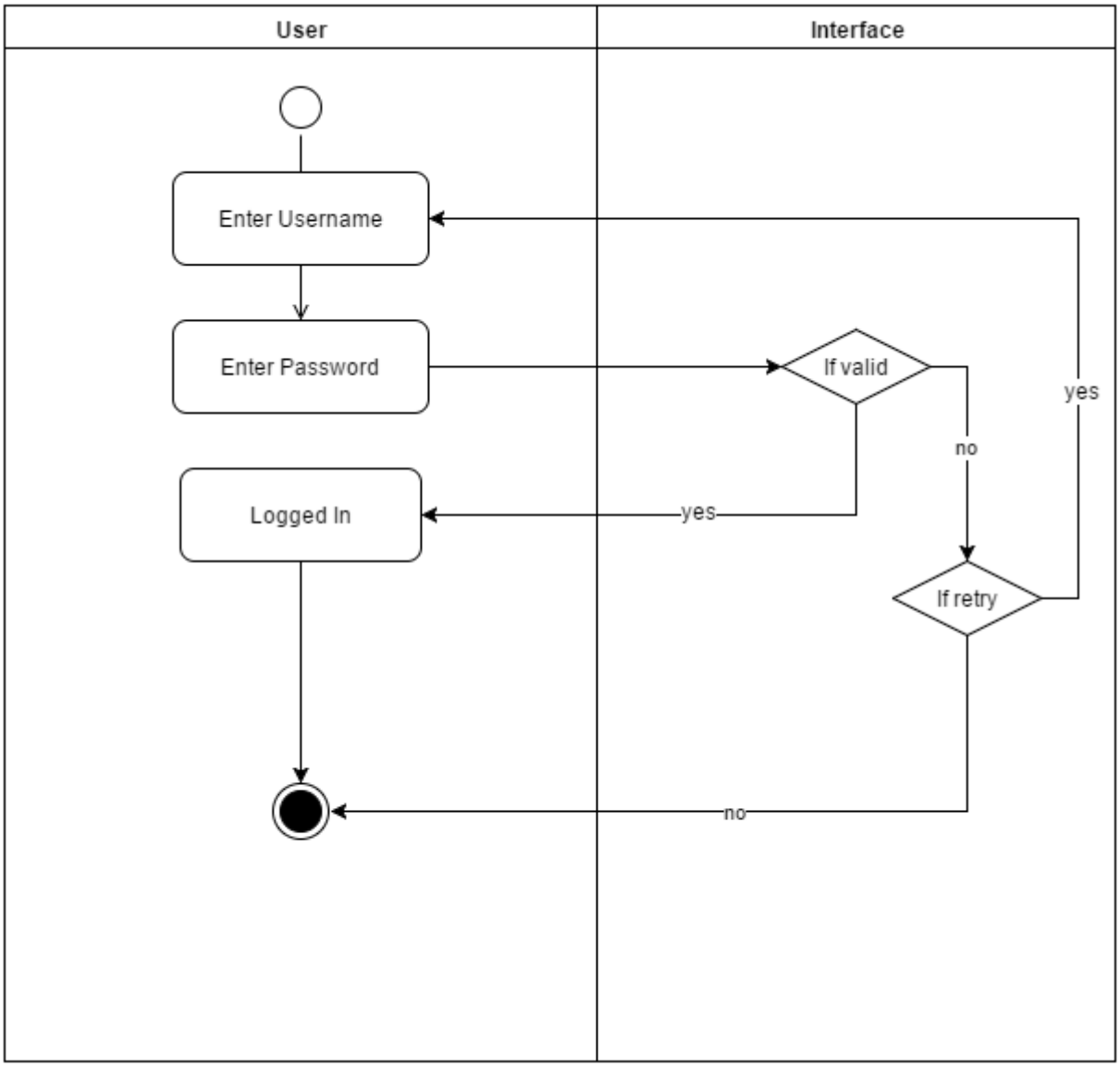


Figure: Swim lane diagram for user login

4.3.2.3 Swim lane diagram for logout

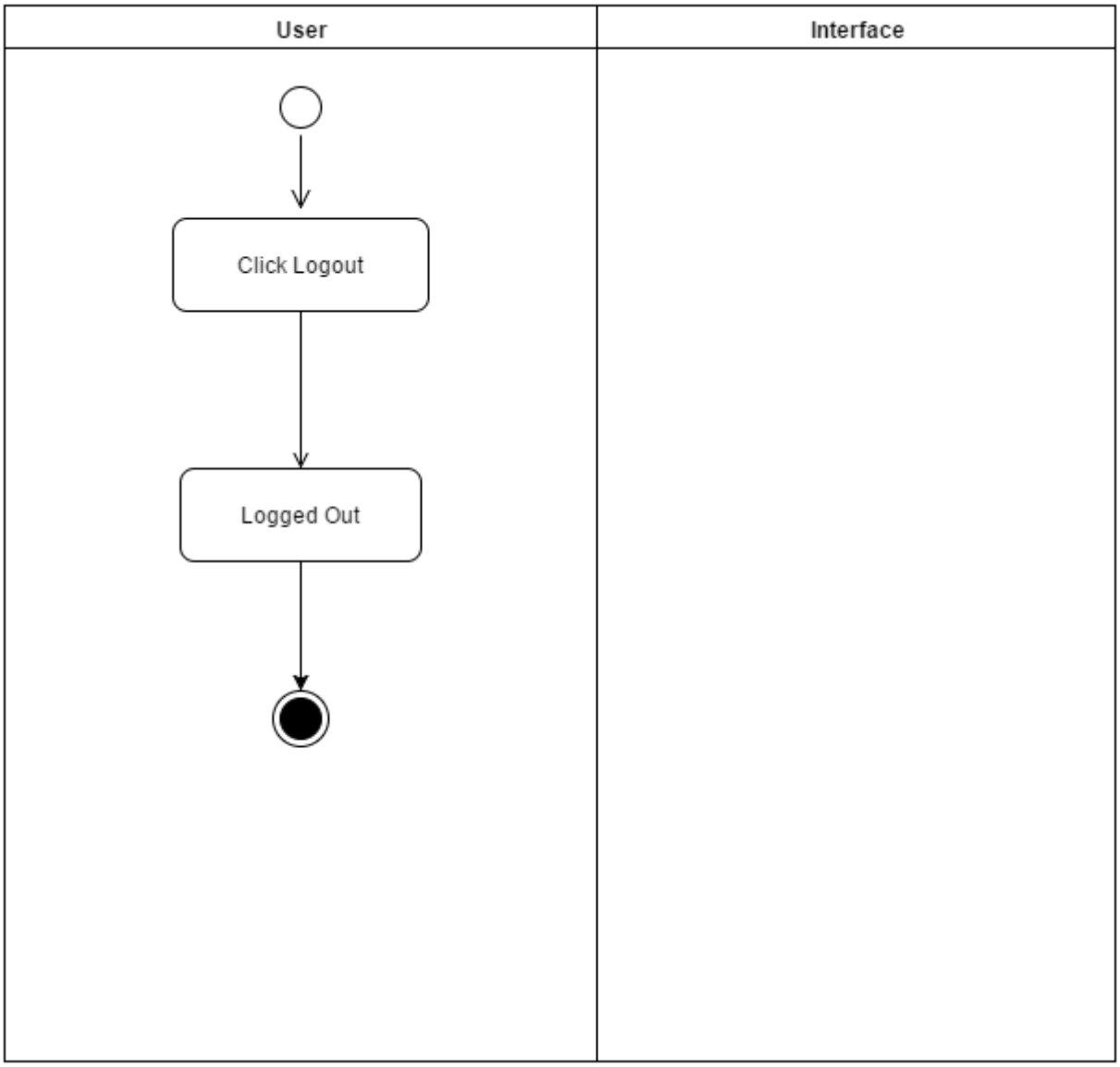


Figure: Swim lane diagram for logout

4.3.2.4 Swim lane diagram for seat allocation

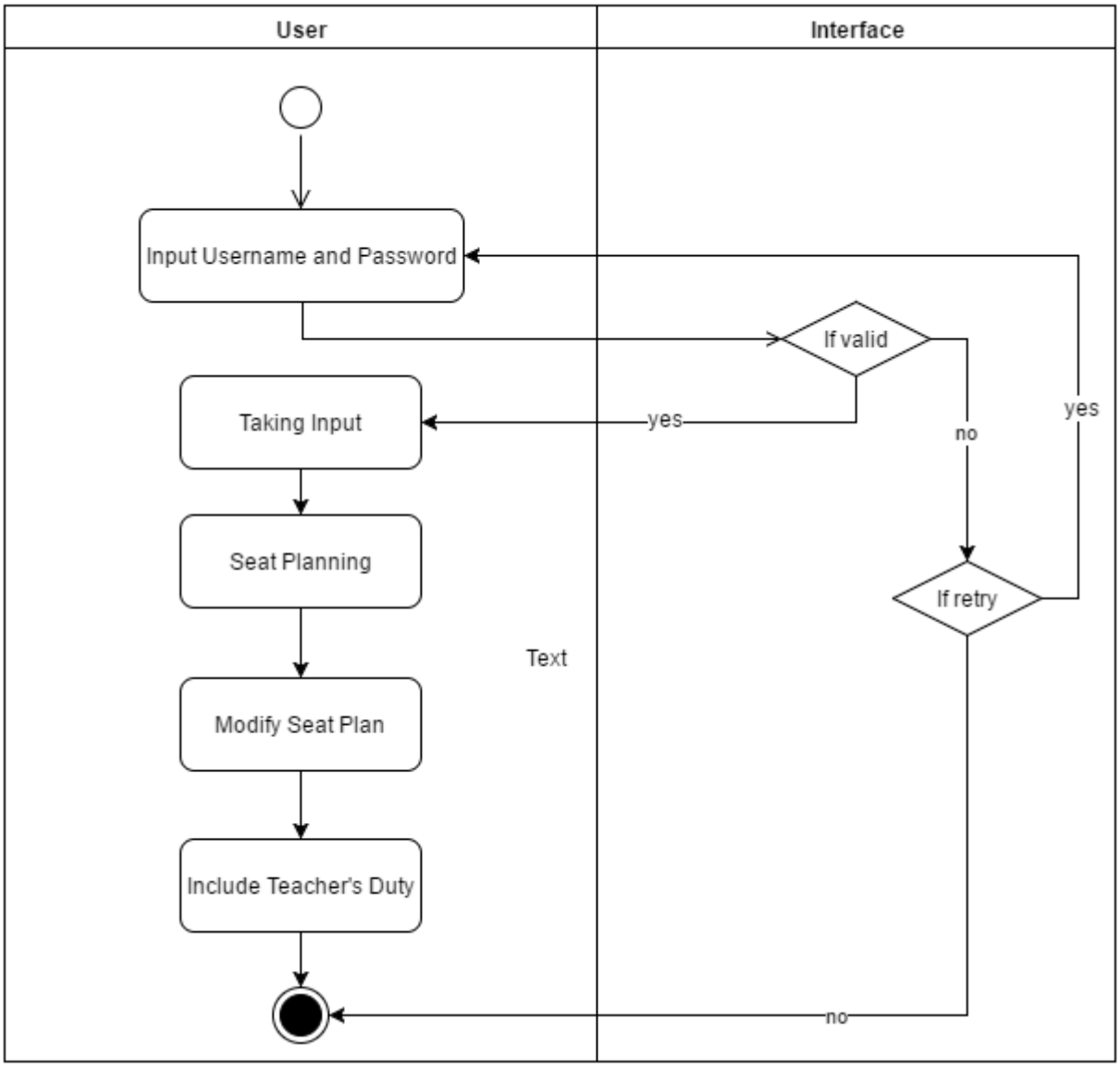


Figure: Swim lane diagram for seat allocation

4.3.2.5 Swim lane diagram for system output

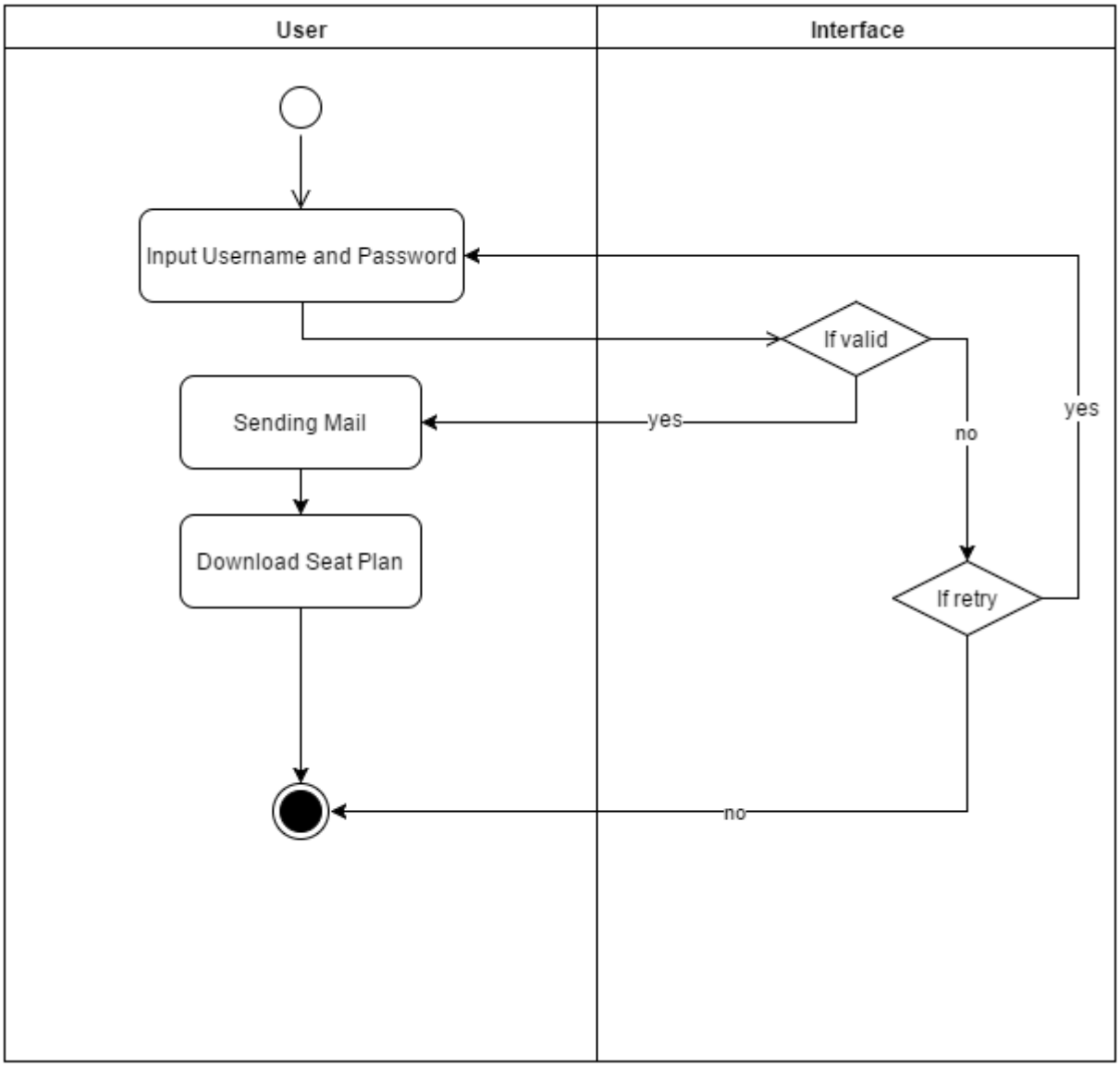


Figure: Swim lane diagram for system output

Chapter 5: Data Model

5.1 Data modeling concepts

In cases where software requirements include the need to create, extend, or interface with a database or if complex data structures must be constructed and manipulated, the software team may choose to create a data model as part of overall requirements modeling.

5.2 Data objects identification

A data object is a representation of composite information. Composite information means the data objects will have multiple attributes. We need to identify those data objects. We used the "grammar parsing" method for this purpose, that is, we selected nouns from the scenario and searched which one has multiple attributes in our context. A table in this regard is given below-

Noun	**Prob./Sol. space**	**Attributes**	**Decision**
User	Sol. space	Email address, password	Accepted
User email	Sol. space	n/a	Attribute of user
Password	Sol. space	n/a	Attribute of user
Registration	Prob. space	n/a	n/a
Link	Prob. space	n/a	n/a
Student's roll num.	Sol. space	n/a	Attribute of seat
Noun	**Prob./Sol. space**	**Attributes**	**Decision**
Student's email	Sol. space	n/a	Attribute of seat
Room Plan	Sol. Space	Room number, Max columns, Max rows	Accepted

Room number	Sol. space	n/a	Attribute room plan
Max columns	Sol. space	n/a	Attribute of room plan
Max rows	Sol. space	n/a	Attribute of room plan
Seat	Sol. space	Student roll, Student email, Row number ,column Number	Accepted
Teacher	Sol. space	Teacher's name	Accepted
Column	Sol. space		Accepted

5.3 Selected data objects

We found following data objects in our Seat Planning System –

User:
User Id
Email Address
User Name
Password

Room Plan:
Room_Plan_Id
User Id
Room number
Max rows
Max columns

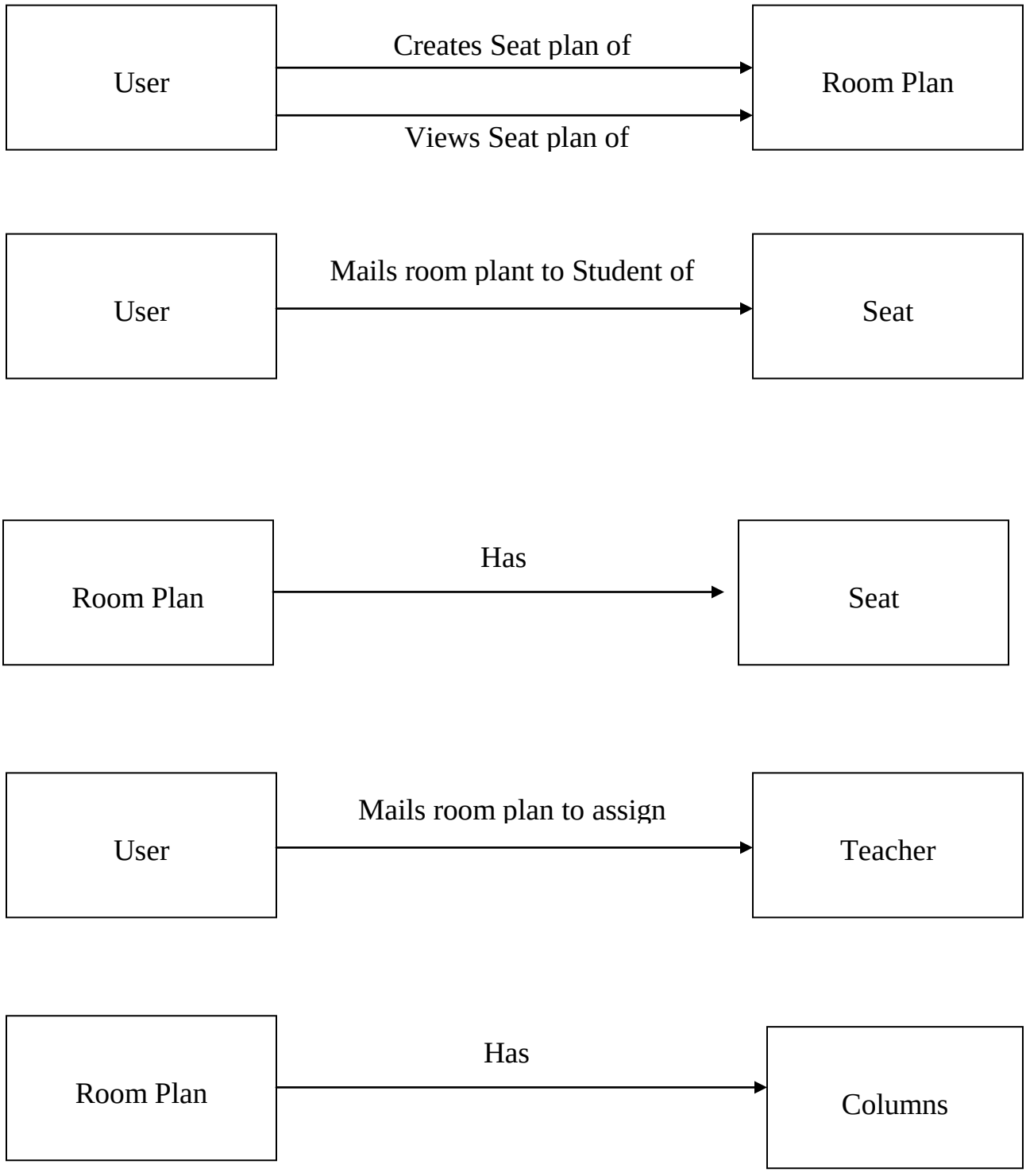
Length of Exceptional Columns
Date of Exam
Seat Planning html/Pdf file

Seat:
Seat_Id
Room_Plan_Id
Column number
Row number
Student email

Teacher
Teacher_Id
Room_Plan_Id
Teachers Name
Teachers Phone Number
Teachers Email

Columns
Column_Id
Room_Plan_Id
Column Position
Length of Column

5.4 Relationship between Data Objects



5.5 Entity Relationship Diagram

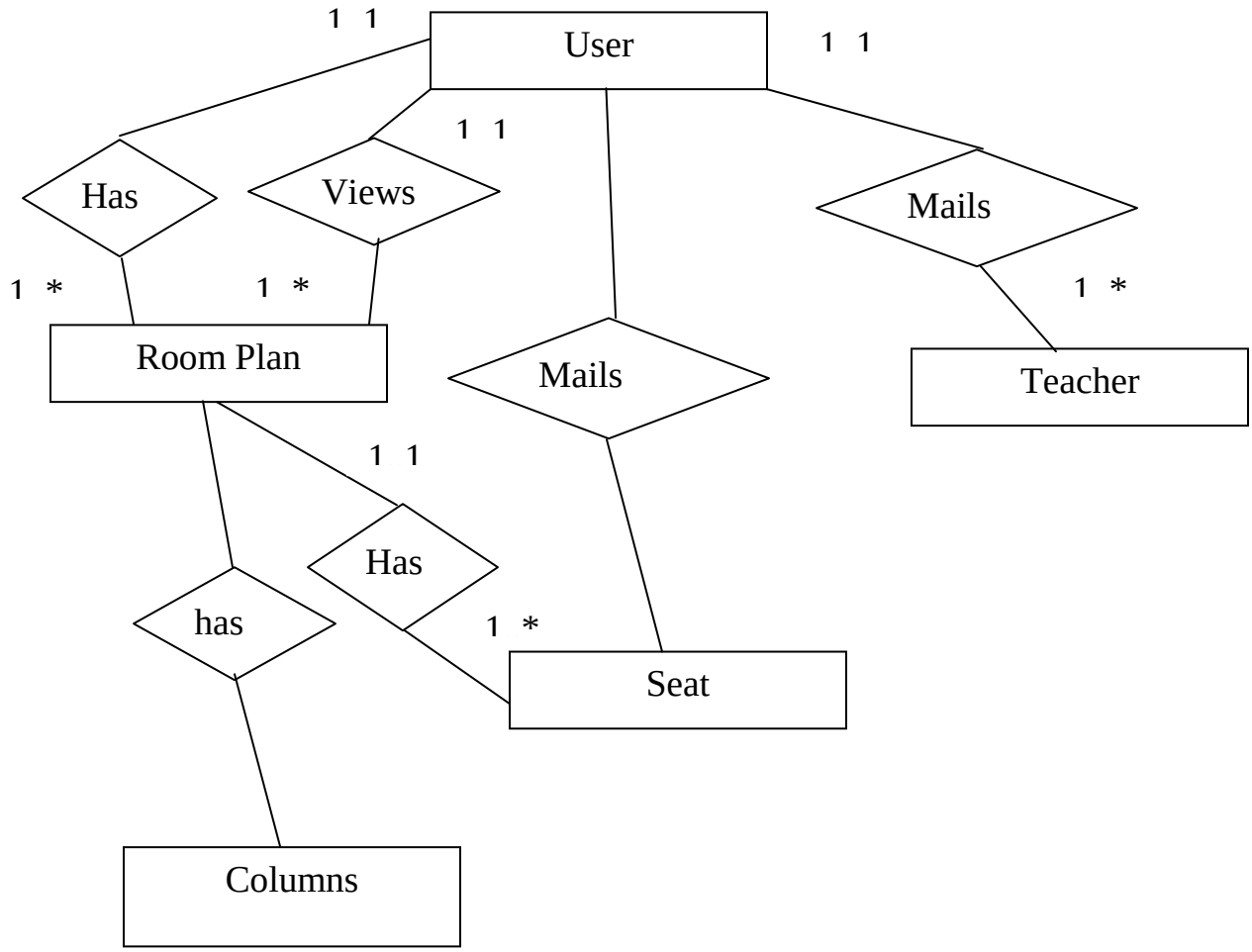


Figure: Entity Relationship diagram

5.6 Schema

User	
Attribute	Data Type
User Id	Int
Email Address	Varchar(100)
User Name	Varchar(100)
Password	Varchar(100)

Room Plan	
Attribute	Data Type
Room_Plan_Id	Int
User Id	Int
Max rows	Int
Max columns	Int
Date of Exam	Date
Seat Planning html/Pdf file	Varchar(100)

Seat	
Attribute	Data Type
Seat_Id	Int
Room_Plan_Id	Int
Column number	Varchar(100)
Row number	Varchar(100)
Student email	Varchar(100)

Teacher	
Attribute	Data Type
Teacher_Id	Int
Room_Plan_Id	Int
Teachers Name	Varchar(100)
Teachers Phone Number	Varchar(100)
Teachers Email	Varchar(100)

Columns	
Attribute	Data Type
Column_Id	Int
Room_Plan_Id	Int
Column Position	Int
Length of Column	Int

Chapter 6: Class Based Model

6.1 Class Based Modeling Concept

Class-based modeling represents the objects that the system will manipulate, the operations that will applied to the objects, relationships between the objects and the collaborations that occur be-tween the classes that are defined.

6.2 Identifying Analysis Classes

Analysis classes can be identified by parsing noun and noun phrases. Then they need to be classified by general classification and selection criteria.

General classification:

1. External entities
2. Things
3. Occurrence or events
4. Roles
5. Organizational units
6. Places
7. Structures

Selection criteria:

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

Potential Classes	General classification	Selection Criteria	Accepted/Rejected
User	Roles	2, 3, 4, 5, 6	Accepted
Students	External entities	1, 3, 4, 6	Accepted
Seat plan	Things	1, 2, 3, 5, 6	Accepted
Email			Rejected
Password			Rejected
Roll number			Rejected
Room	Organizational unit	1, 3, 4, 6	Accepted
Teachers	External entity	1, 6	Rejected

6.2.1 Accepted Classes:

- User
- Students
- Room
- Seat plan generator

6.2.2 Attribute identification:

Class Name	Attributes
User	Email Password
Students	Number of students Roll number Email
Room	Room number Room size Dummy seat
Seat plan	Students Room Teacher'sList

6.2.3 Method Identification:

Verb Identification:

Verb	Remark
Register	Yes
Verify	Yes
Log in	Yes
GenerateSeatPlan	Yes
Take	Out of scope
Randomize	Yes

Mention	Out of scope
Modify	Out of scope
Send	Out of scope

Methods:

Class Name	**Methods**
User	setEmail() setPassword() register() verify() login() seatPlan()
Students	setNumbreOfStudents() getNumberOfStudents() setEmail() getEmail() setRoll() getRoll()
Room	setRoomNumber() getRoomNumber() setRoomSize() getRoomSize() setDummySeat() getUummySeat()
SeatPlan	getStudents() getRoom() generateTeachersList() randomizeRoll() generateSeatPlan()

6.2.4 Class Cards:

User	
Email Password Verification code	setEmail() setPassword() register() verify() login() seatPlan()

Students	
Number of students Roll number Email	setNumbreOfStudents() getNumberOfStudents() setEmail() getEmail() setRoll() getRoll()

Room	
Room number Room size Dummy seat	setRoomNumber() getRoomNumber() setRoomSize() getRoomSize() setDummySeat() getUummySeat()

SeatPlan	
Students Room Teachers	getStudents() getRoom() generateTeachersList() randomizeRoll() generateSeatPlan()

6.3 Class Responsibility Collaboration (CRC) Diagram

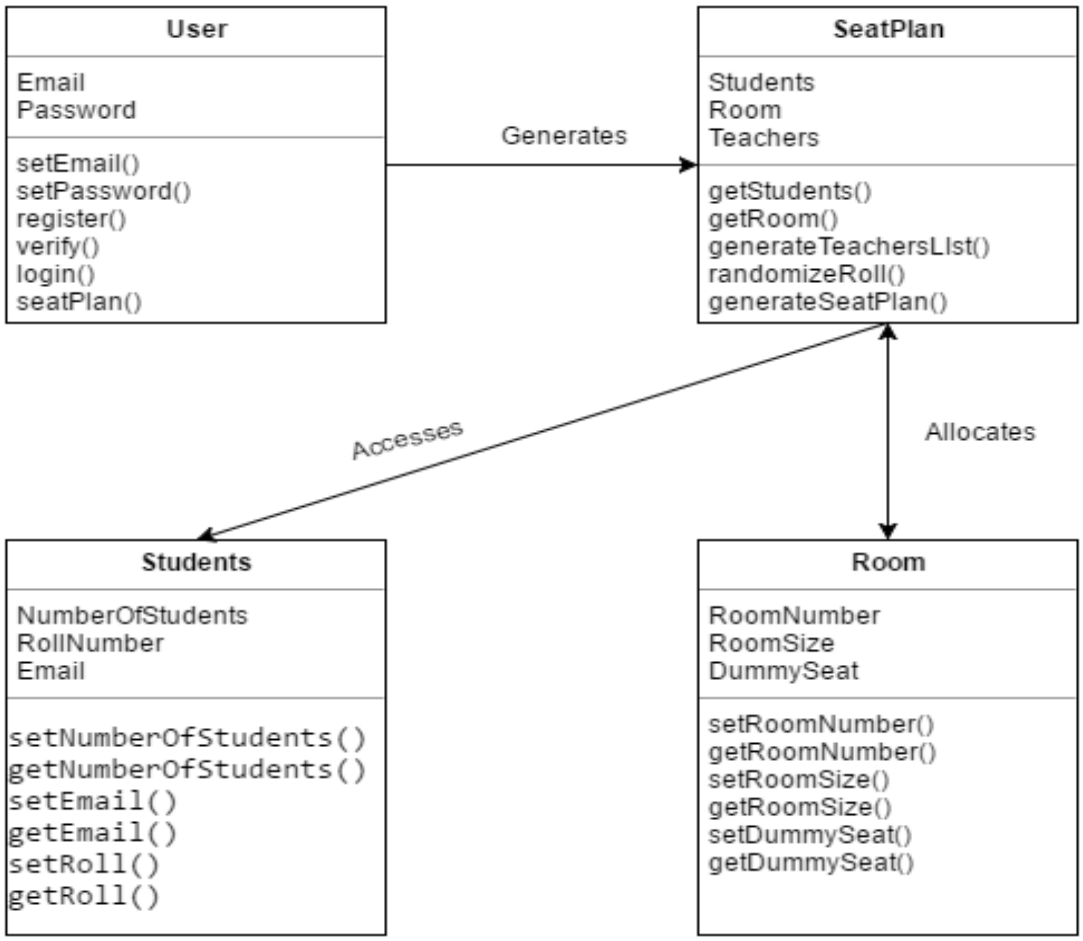


Figure 6.1: CRC Diagram

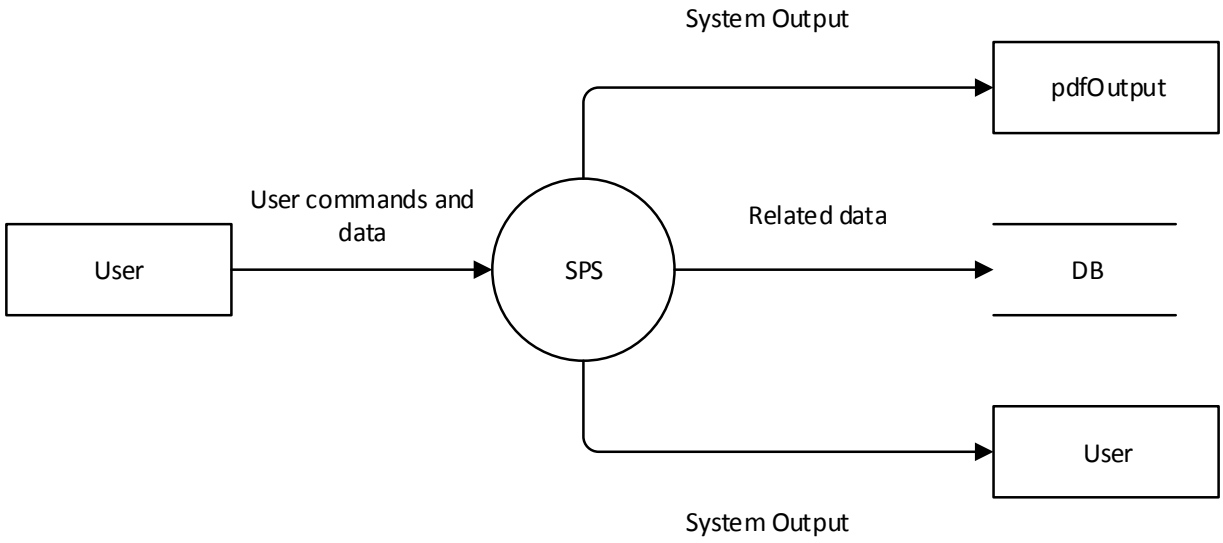
Chapter 7: Flow Oriented Model

7.1 Introduction

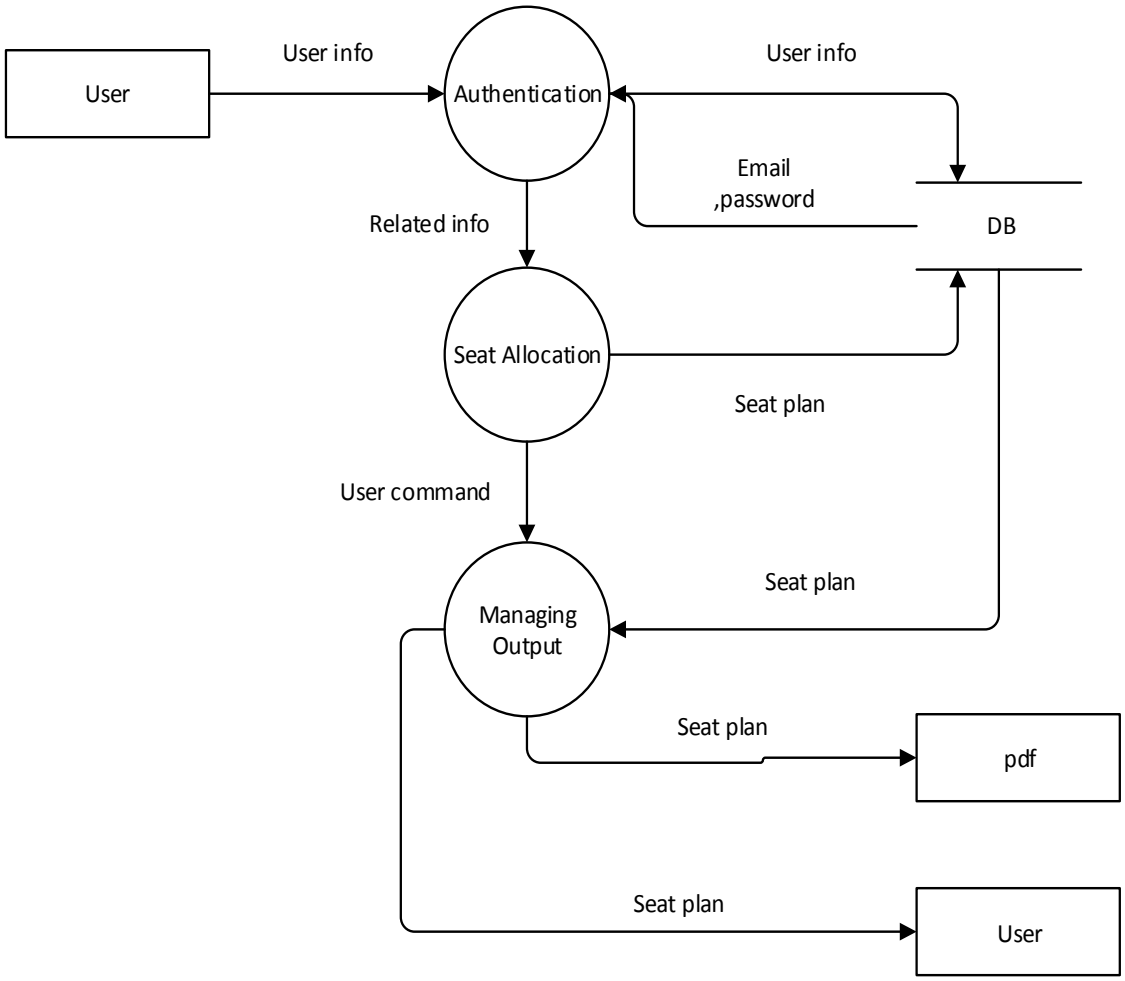
Although data flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirements analysis notations in use today.

7.2 Data Flow Diagram (DFD)

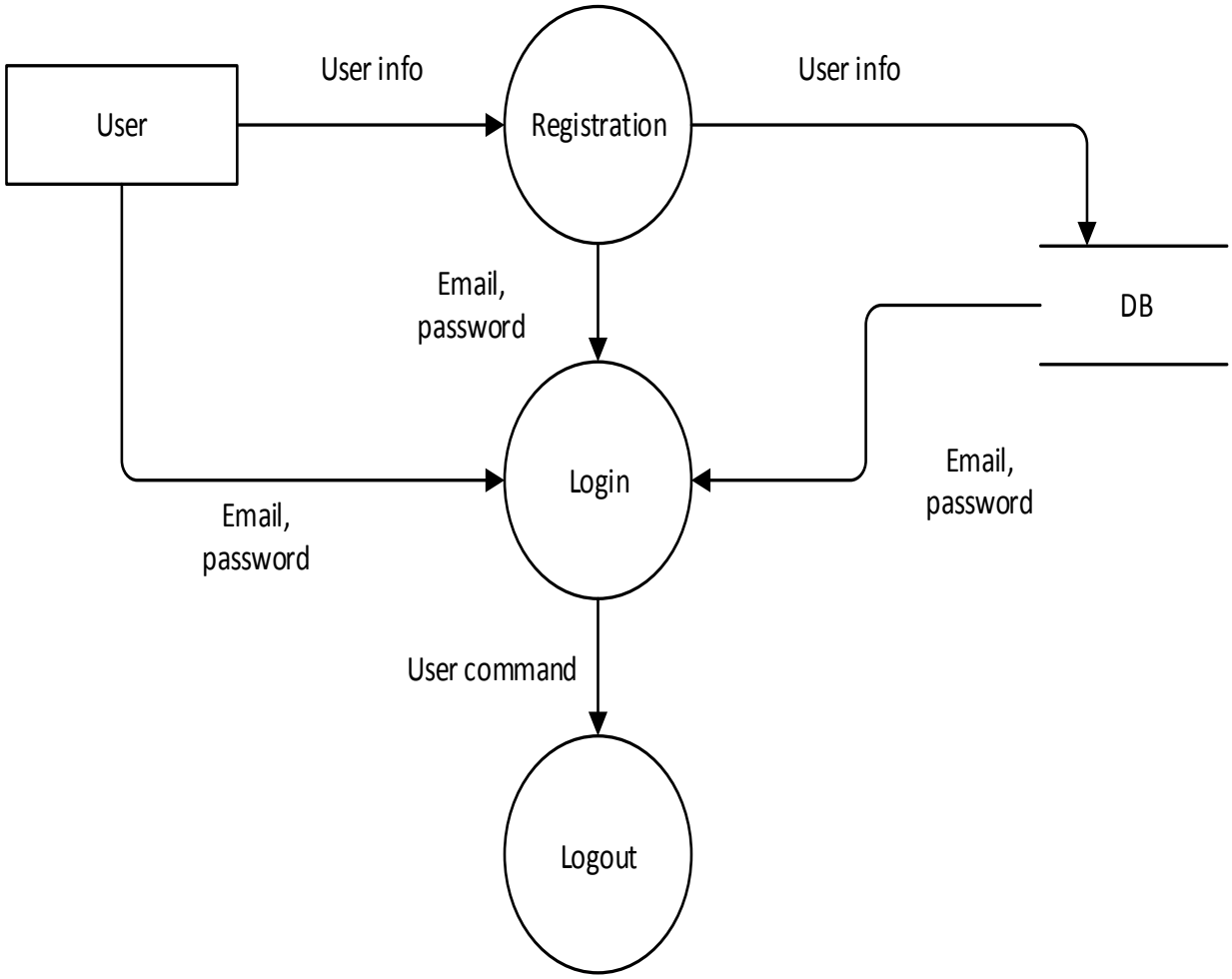
The Data Flow Diagram (DFD) takes an input-process-output view of a system. Data objects flow into the software, are transformed by processing elements and resultant data objects flow out of the software. Data objects are represented by labeled arrows and transformations are represented by circles.



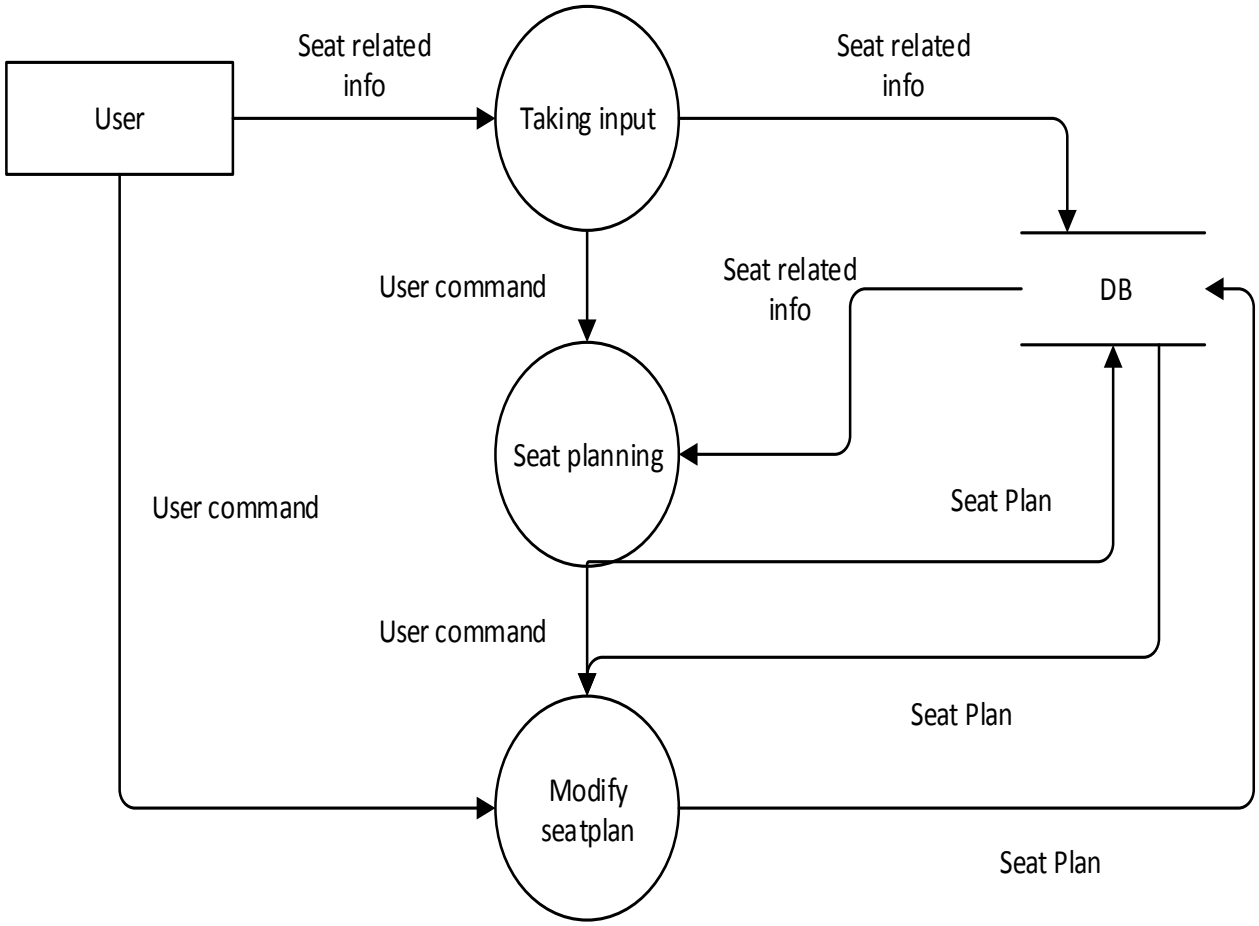
Level 0: DFD for Seat planning system



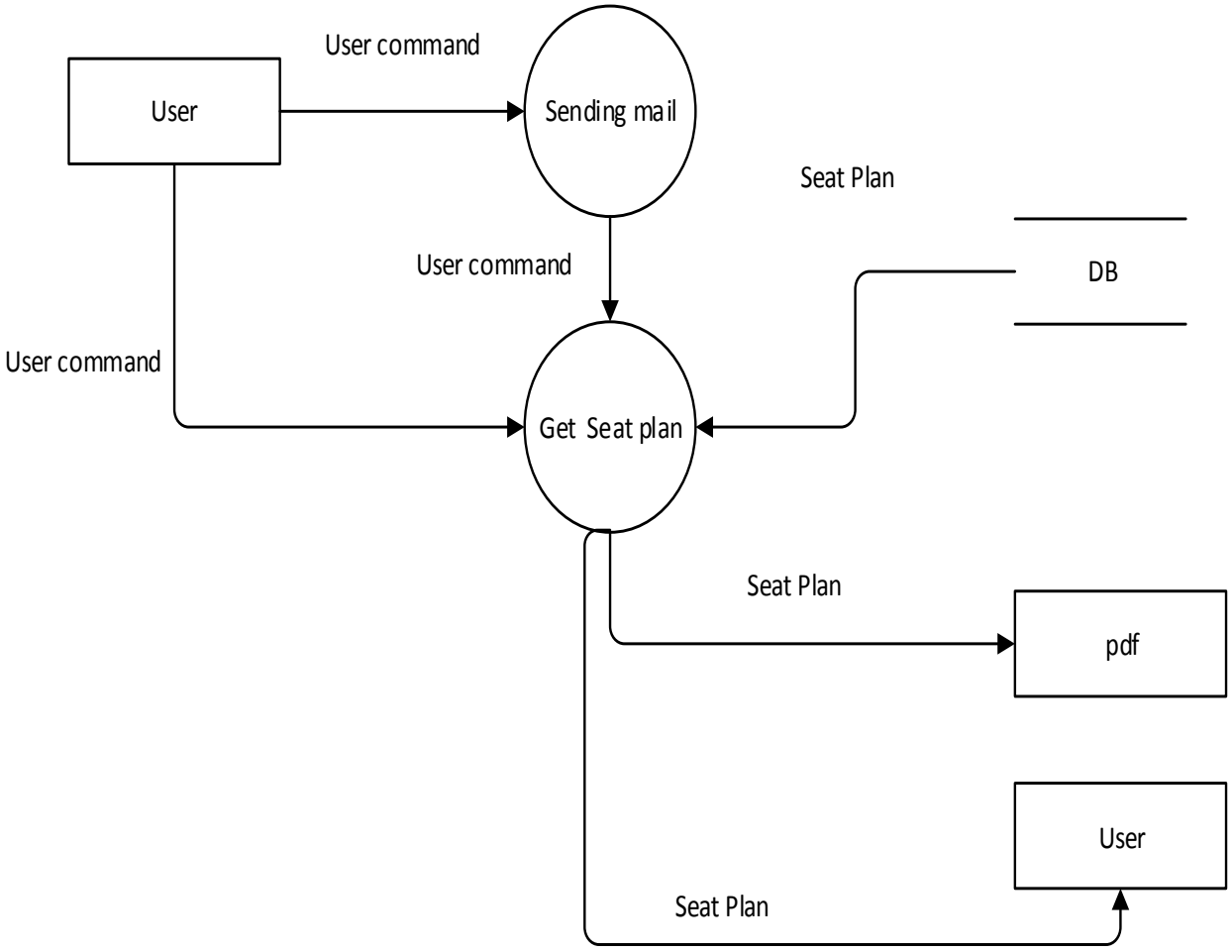
Level 1: DFD for Seat planning system



Level 2.1: DFD for Seat planning system (Authentication)



Level 2.2: DFD for Seat planning system (Seat Allocation)



Level 2.3: DFD for Seat planning system (Managing Output)

Chapter 8 Behavioral Model

8.1 Introduction

Behavior modeling is also referred to as State modeling, State machines and State transition matrix. Behavior modeling is when one thinks of his ideas in terms of states and transitions. This requires both identifying all of the interesting states of being that software or its components are likely to be in. And also, at a high level abstracting what events are likely to cause software or its components to change between states of being.

8.2 Identifying Events

Here we have identified events from the Usage Scenario and listed their corresponding initiators & collaborators.

Count	Events	Initiator	Collaborator
1	Registration	User	
2	Login	User	
3	Verification	User	
4	Seat plan generation	User	SeatPlan
5	Setting information	Students, Room	
6	Getting information	Students, Room	SeatPlan
7	Placing Dummy seat	SeatPlan	
8	Randomizing rolls	SeatPlan	
9	Sending email	SeatPlan	

8.3 State Transition Diagram

State Transition Diagram represents active states for each class and the events (triggers) that cau-se changes between these active states. Here we have provided diagram for each of the actors.

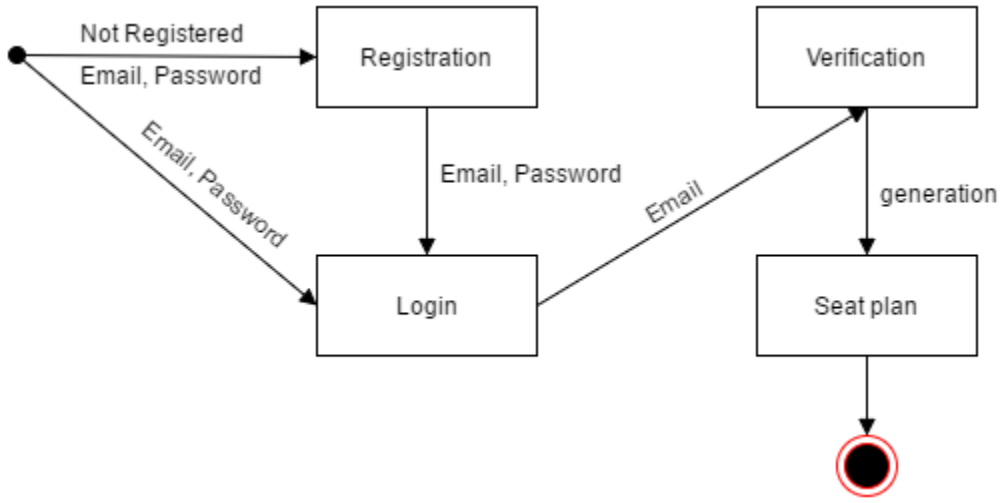


Figure 8.1: State transition diagram of user

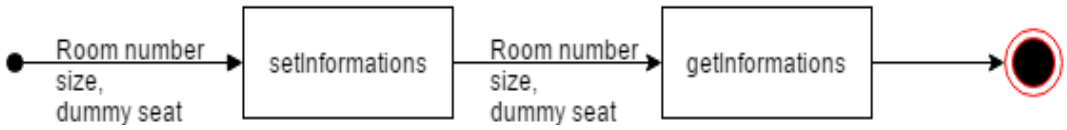


Figure 8.2: State transition diagram of Room

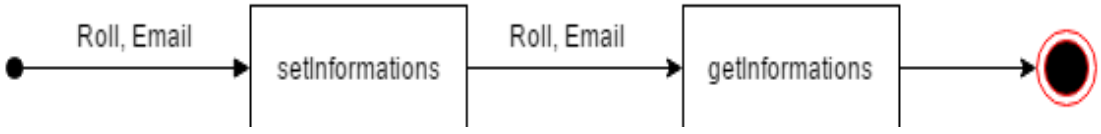


Figure 8.3: State transition diagram of student

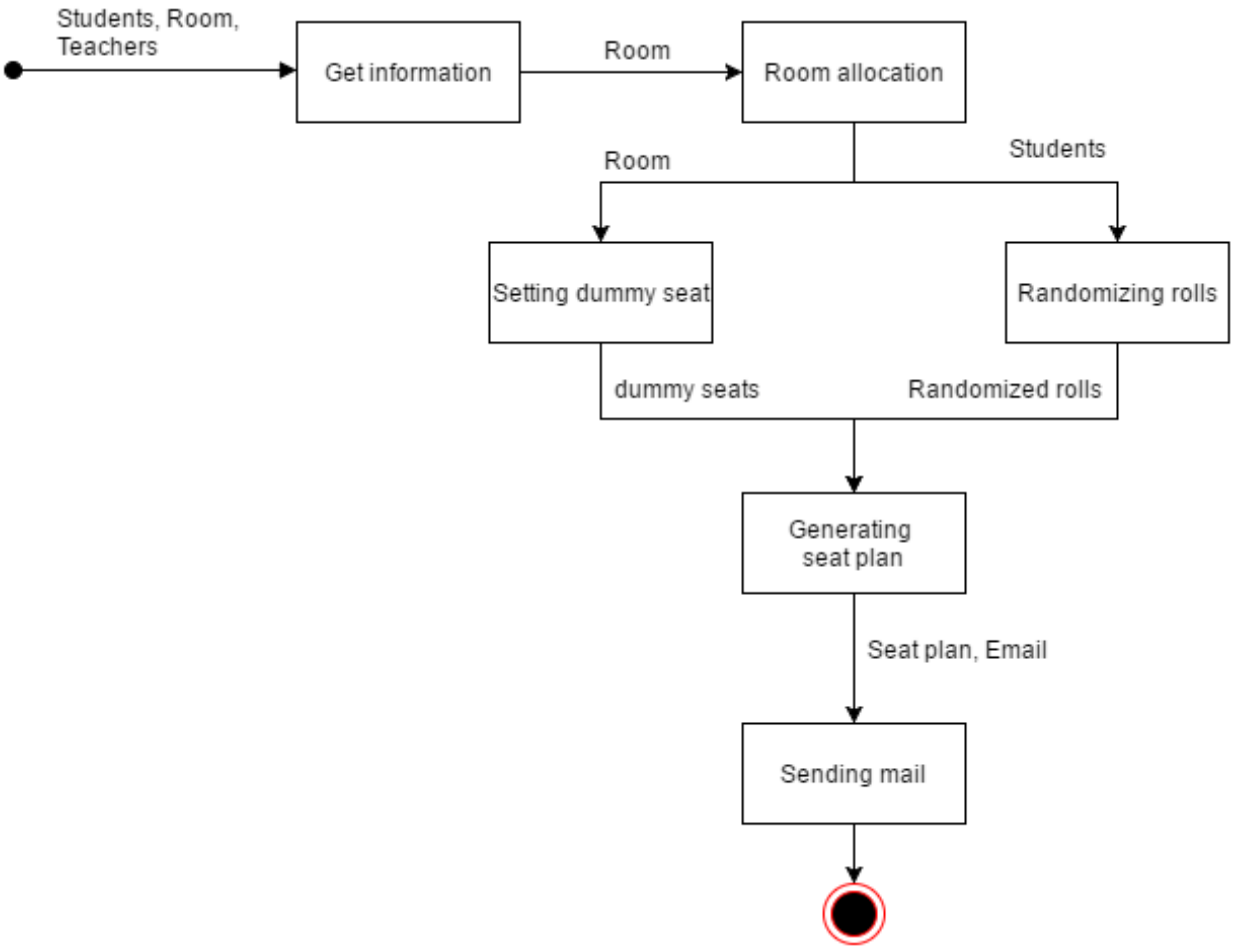


Figure 8.4: State transition diagram of SeatPlan

8.4 Sequence Diagram

Sequence Diagram indicates how events cause transitions from object to object. It is actually a representation of how events cause flow from one object to another as a function of time.

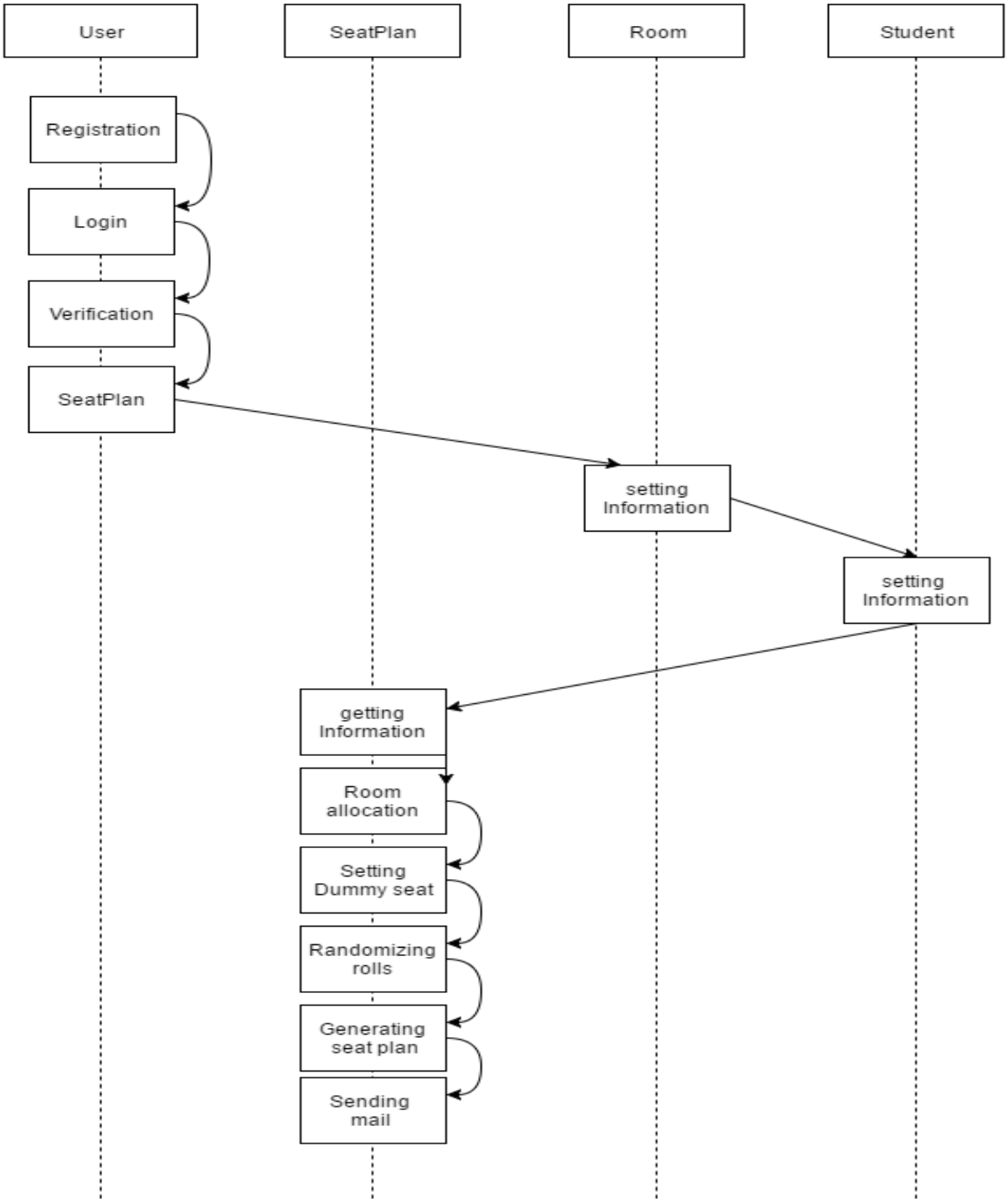


Figure 8.5: Sequence diagram

Chapter 9

9.1 Conclusion

It is a pleasure to submit the final SRS report on “Seat Planning System”. From this, the user will get a lot of advantages about exam seat planning.

9.2 References

1. Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th Ed.). Boston, Mass: McGraw-Hill. ISBN 0-07-285318-2.

2. Swim lane Diagram
   https://www.smartdraw.com/swim-lane-diagram

3. Visio drawing
   https://www.google.com/search?q=how+to+add+arrow+in+visio&ie=utf-8&oe=utf-8

4. E-R Diagram
   http://www.codeproject.com/Questions/439447

5. Data flow diagram
   http://www.codeproject.com/Questions/143117

6. Behavior-Modeling-Lesson
   http://www.codeproject.com/Articles/6675

7. Sequence Diagram
   https://www.google.com/search?q=codeproject.com+sequence+diagram&source=lnms&tbm=isch&sa=X&ved=0ahUKEwipme7tionMAhUO8mMKHSstBZwQ_AUIBygB&biw=1366&bih=657#imgrc=HlO_RZF_qGWZDM%3A