

Syracuse University

AttendU: An Online Attendance Check-In Platform

Software Requirements Specification

Team Members:

Ruxin Wang

Mingyuan Xie

Minrui Zhang

Qinzhe Zhang

Instructor:

Edmund S. Yu

December 10, 2018

Submitted in partial fulfillment

Of the requirements of

CSE 682 Software Engineering

Contents

Preface	4
I. Introduction	5
1. Purpose.....	5
2. Scope of Project	5
3. Overview of Document	5
II.Glossary	7
III.User Requirements	7
1. Functional User Requirements.....	8
1.1.Product Perspective	8
1.2.Product Functions.....	8
1.3.User Characteristics.....	9
2. Non-Functional User Requirements.....	10
2.1.Operating Environment.....	10
2.2.Design and Implementation Constrains	10
2.3.Assumptions and Dependencies	10
IV.System Architecture.....	10
1. Client End	11
AngularJS	11
2. Server End	12
Spring Boot	12
Microservices.....	12
MyBatis	12
MySQL 12	
V. System Requirements Specification	13
1. Functional Requirements	13
1.1.Homepage	13
1.2.User Login	15
1.3.User Register.....	15
1.4.Dashboard	16
1.5.Check In	20
2. Non-Functional Requirements	21
2.1.Performance requirements	21
2.2.Security requirements	21
2.3.Reliability requirements.....	21
2.4.Compatibility requirements	22

2.5. Usability requirements	22
2.6. Maintainability requirements.....	22
VI. System Models	23
1. Use case Diagram	23
2. Activity Diagram.....	25
3. Sequence Diagram	27
4. Class Diagram	29
VII. References.....	30

Preface

This Software Requirements Specification (SRS) is intended for both users of the Web-based Attendance Management System and administrators evaluating its potential use. It is also useful to any maintainers of the interface's codebase. Version One of the SRS covers the interface at the time of Release Two, which occurred at 9th Week of the Design Process.

I. Introduction

1. Purpose

In most institutions, attendance is taken manually. It is time-consuming, unreliable and even insecure. Web-based Attendance Management System can be a better choice to replace the paper-based remises of attendance management system. This new system can now handle large number of participants and events over large networks. Web-based Attendance Management System is easy to use, while being a very powerful attendance management system that can cover schools, corporations, forums, and activities etc.

2. Scope of Project

The main characteristics of this system is web based, fully responsible and flexible. The purpose of making a Web-based Attendance Management System is to manage attendance information in all kinds of events. The system can record attendance information including attendance time, departure time, and attendance status etc.

To simplify usage, users have two choices to enter this system and record attendance log

- They can type activity numbers and then go to attendance checking activity directly.
- They can log in and then go to the dashboard, users can view and manage their subscribed events.

Room and activity configurations are important to help our system record attendance. Room is for hosting a series of activities, for instance, a room represents a class. Every room can host many activities for participants in the room, for instance, an activity stands for the lecture attendance of October 12th. Initially, organizer or authorized administrators can define details for the room, such as the time limitation, participant identities requirement and default configuration for activities. For each activity, administrators can re-configure the setting or remain default. The organizer can check the attendance overview any time for activities. Moreover, the organizer or authorized users can view statistical results from attendance data. The visualization of statistical results and data are one of the key features for this application.

There are other features can benefit users. For example, users can report absence for an activity in advance. The activity organizer also can get absence information before the start time. Users can also upload evidences for absence.

3. Overview of Document

The User Requirement part, of this document are divided into two sections ---- Functional User requirements and Non-functional User Requirements. This part describes the informal requirements and is used to establish a context for the technical requirements specification.

The Glossary part, of this document defines the technical terms used in the document.

The System Architecture part, of this document present a high-level overview of the anticipated system architecture.

The Requirements Specification part, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product, showing the distribution of functions across system modules.

The System Models part, of this document includes graphical system models showing the relationships between the system components and the system and its environment.

Both parts of the document describe the same software product in its entirety but are intended for different audiences and thus use different language.

II. Glossary

Term	Definition
Organizer	The user who defines details for the room and activity, such as the time limitation, participant identities requirement and default configuration for activities.
Attendee	The user who needs to help recording their attendance status via this application.
Admin	System administrator who has special permission to manage and administer/control the system
Room	Place where for hosting a series of activities, for instance, a room represents a class.
Backend	Mainly represents the application logic responsible for handling request-response transactions.
Middleware	Represents a function that is invoked by the routing layer (Restful APIs) before the final request handler.
Database	Denotes MySQL, a database for storing registration information, Room information and related activities information.
Activity	Every room can host many activities for participants, for instance, an activity stands for the lecture attendance of September 12th.
Scenarios	The tags of the attendance events. (e.g. School, Business...)
Software Requirements Specification	A document that completely describes all functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.
User	Organizers or Attendees.

III. User Requirements

This section will give an overview of the whole system. The system will be explained in its context to show how different components interact and affect each other. It will also define use cases for the application, offer a clear, yet concise explanation of the services offered

with the help of system models, and different phases of client-server interaction through sequence diagrams.

1. Functional User Requirements

1.1. Product Perspective

AttendU is a new self-contained web application. As with all web applications, it adopts a client-server model for authentication, authorization and communication. The goal of the application is to make the process of attendance taken easy and hassle-free. In most institutions, attendance is taken manually. It is time-consuming, unreliable and even insecure. AttendU can be a better choice to replace the paper-based premises of attendance management system. This new application can now handle large number of participants and events over large networks.

The client part of the application is constituted by the organizers. The application logic is implemented with Angular.js as the web framework, built on top of the Bootstrap JavaScript designing framework. The primary database is MySQL. This is used for storing data such as user data, room data, and activity data. The crux of the application is a convenient and flexible attendance taken system. Room and activity configurations are important to help our system record attendance. Room is for hosting a series of activities, for instance, a room represents a class. Every room can host many activities for participants in the room, for instance, an activity stands for the lecture attendance of October 12th. Initially, organizer or authorized administrators can define details for the room, such as the time limitation, participant identities requirement and default configuration for activities. For each activity, administrators can re-configure the setting or remain default. The organizer can check the attendance overview any time for activities. Moreover, the organizer or authorized users can view statistical results from attendance data. The visualization of statistical results and data are one of the key features for this application. There are other features can benefit users. For example, users can report absence for an activity in advance. The activity organizer also can get absence information before the start time. Users can also upload evidences for absence.

1.2. Product Functions

The product aims to mitigate the trouble that attendance taken is time-consuming, unreliable and even insecure. Each of the product's features will be elaborated in the coming sections. However, a bigger picture of some of the most important functions that the product must incorporate is essential to understanding how the product is expected to behave. Some of the important functions of the product are described below,

- UR1: The system must allow users to have access to a clean and responsive user interface, courtesy of assimilating the best design practices.
- UR2: It must provide a strong and robust authentication and authorization mechanism since users trust the application with personal data.
- UR3: The system must have efficient lookup algorithms for accurate detection and mapping of specific rooms or activities.
- UR4: The system must provide flexibility for organizers to define details for the room, such as the time limitation, participant identities requirement and default configuration for activities.

- UR5: Flexibility for attendees need to help recording their attendance status.

1.3.User Characteristics

The application will consist of 2 user flows, one for the organizers of this application, and the other for the attendees. A high-level summary of the most important features offered by the system with respect to each of these users are as follows,

- For organizers: They can define details for the room, such as the time limitation, participant identities requirement and default configuration for activities. The organizer can also check the attendance overview any time for activities. Moreover, the organizer or authorized users can view statistical results from attendance data.
- For attendees: If the room is set to be anonymous, users simply needs to enter the room number and then doing attendance check. Otherwise, they required to log in first, and then go to dashboard to find their related activities and then click in correspond room to do attendance check.

A high-level diagrammatic representation of the user activity is depicted below,

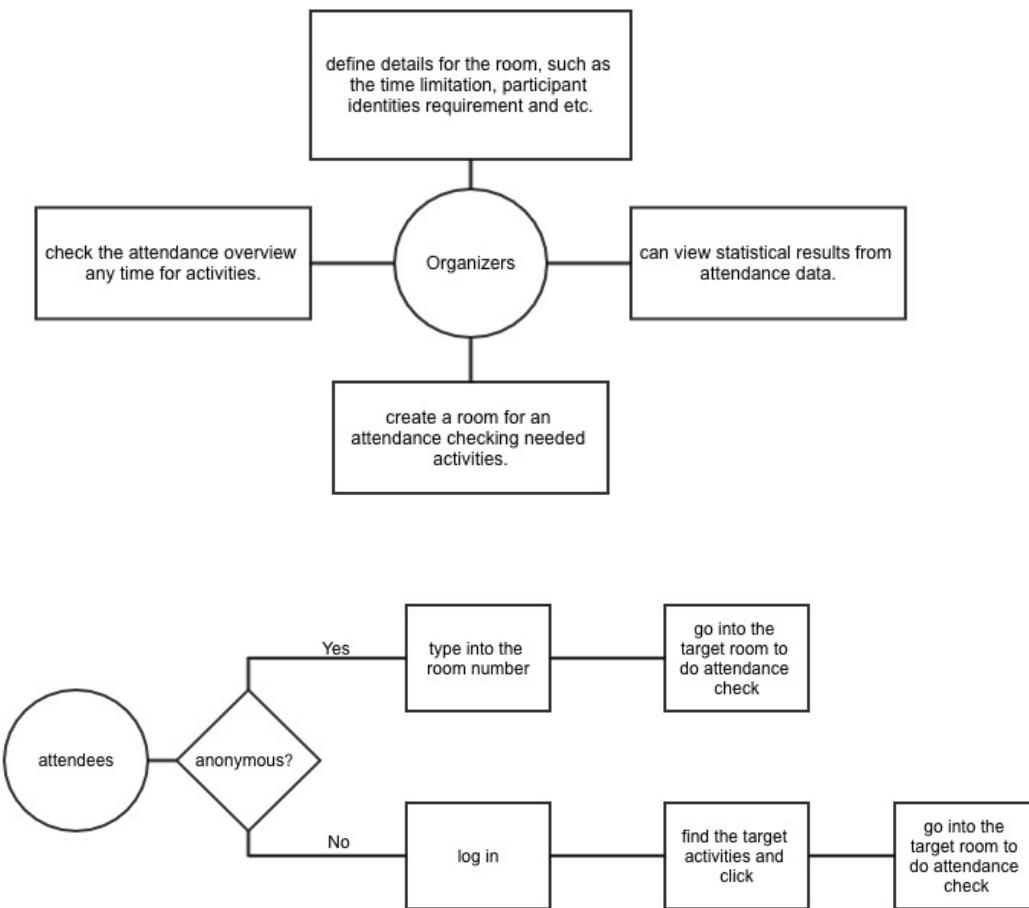


Figure 1 A high level diagrammatic representation of the user activity

2. Non-Functional User Requirements

2.1. Operating Environment

The system will implement the model-view-controller software architecture pattern. The operating environment for the system consists of a browser that renders web pages, an external web server that handles user request-response transactions, an authentication service and a database for storing application data. Servers with reasonable load handling capacity will be enough for web hosting and data storage. Central administration is vested in the web/application server.

2.2. Design and Implementation Constraints

Some of the key constraints associated with the product and its environment are:

- One of the key constraints of the system is the inherent reliability on an internet connection. Since the product is implemented using the client-server model, it is imperative to have an active internet connection.
- Another important constraint is that the organizers must store the application needed information (e.g. attendees and activities information) as accurately and completely as possible.
- Administrative responsibility is relegated to the PaaS vendor. As a result, managing the application becomes a little tricky for the local administrative team.
- Developing and handling a strong and robust authentication and authorization mechanism may turn out to be tricky.

2.3. Assumptions and Dependencies

- Assumptions: Since the system will be deployed using a cloud-based hosting service, an assumption about its continuous, uninterrupted availability is made. The system will depend on how completely and accurately needed information in the process of finishing all functions. Hence, an assumption about total information is made.
- Dependencies: The system will rely heavily on frameworks for user interface development, routing and data handling.

IV. System Architecture

To meet user requirement, a web-based service design has been proposed. The Service contains two parts to provide full service. The first part is client end. To implement light weight and highly efficient solution, our team decided to use a single page web application. The second part is server end. A matured industrial framework would help this service platform be development in quick and high quality.

With two ends service design, the high-level service architecture design can be planned as a big picture as follow figure.

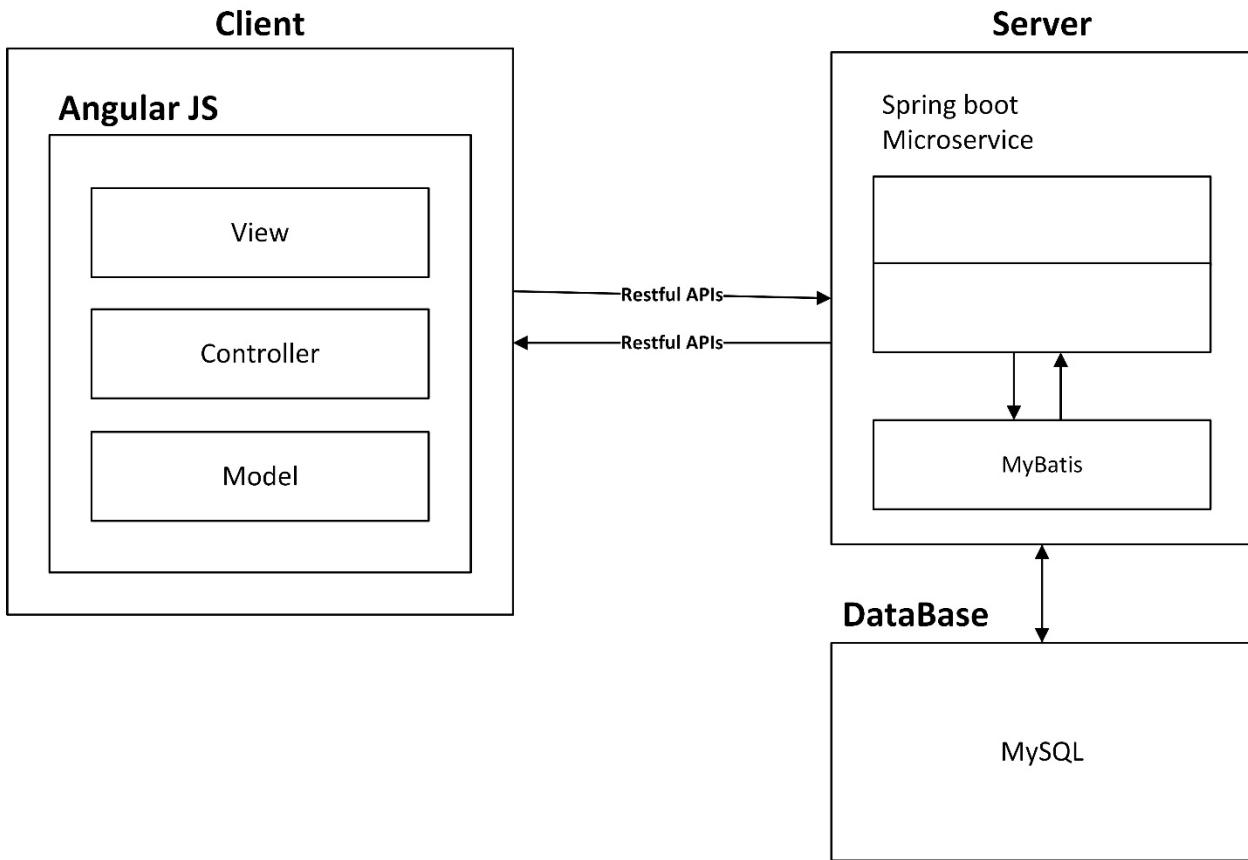


Figure 2 General Architecture

1. Client End

AngularJS

This service platform would use AngularJS as the front-end framework. AngularJS is a JavaScript-based web application framework, and it is good at developing single-page applications. This framework can provide client-side model-view-controller (MVC) architectures. This framework is very famous and has been adopted in many famous projects. AngularJS is very reliable and maintainable, as it is widely used. The framework provides many convenient features, such as page router, cookies management and data binding. To deliver service to users, this framework would host three main components.

View

View component is the front interface interacting with users directly. This component is composed by multiple DOMs. Because the platform is based on the single page application, the main page DOM is the single page. Rest of DOMs are template of parts in the main page, but they will host views for many different features and interfaces. Many AngularJS directives would be embedded into those DOM, to do data binding and controller binding.

Model

Model component is a data mapping component. This part will consist of JavaScript files. Those model files would help front-end application pass data from user and view component

to server end. Most work in model component are manipulating data and handle data communication to server with developed Restful APIs.

Controller

To link data presentation and data packaging between View and Model, controller component works as a vital component. It controls all data binding, and interface presentation regulations. It will guide how to display data, and pass them to DOM. It also packages data and pass them into corresponding models to communicate with server.

2. Server End

Spring Boot

Spring Boot is a Spring framework module, and it provides many features about web applications server. With Spring Boot, the development process could be easier and much more efficient, and people can create high performing, easily testable, and reusable code relying on it. Spring Boot is one of the oldest frameworks in Java, and it has become the most popular Java web application framework. It supports all the leading front-end technology such as AngularJS.

Microservices

Based on the Spring Boot framework, several standalone microservices work currently to provide different services to clients. In a microservices architecture, services are fine-grained. The benefit of decomposing an application into different smaller services is that it improves modularity and makes the application easier to develop and test and more resilient to architecture erosion. Even one of the microservices has been crashed will not impact the whole service system. Moreover, Microservices-based architecture help our team develop this term project with continuous delivery and deployment. To fulfill user requirements, four services proposed.

- User Service: provides user information and validation features for users and the system
- Check in Service: provides attendance check in features to users.
- Room Service: provides room to specific groups or organizations.
- Activity Service: authorized user can host events or activities in a specific room to record attendance.
- Data Service: provides features to help front-end system pull over analyzed data information

MyBatis

MyBatis is a Java persistence framework deal with SQL statements using an XML. It can translate SQL statement into Java methods and accommodate data into customized objects. It helps us interact with MySQL data server in quick.

MySQL

MySQL database is one of the most popular database of relational database management system. It is easy to use and manage. In this project, this database management system would help create data relation and maintain data.

V. System Requirements Specification

The purpose of this section is to describe the behaviors and all kinds of system functionalities of the system.

In general, system requirement has been concluded as following perspectives.

- SR1.1: user interface has index page
- SR1.2: user interface has a dashboard page after login
- SR1.3: user interface has a login page and registration page
- SR1.4: user interface has a page to show involved rooms and activities
- SR1.5: server provides APIs to respond front-end requests
- SR2.1: user interface uses cookie and token authentication mechanism to secure user session
- SR2.2: server uses exception catch mechanism to maintain the stability
- SR3.1: server can manipulate database as transaction
- SR3.2: server provides APIs to respond front-end requests
- SR4.1: user interface has a page to config room and activity configurations
- SR4.2: server provides APIs to save and load configuration of rooms and activities
- SR5.1: user interface has a page to let user check in
- SR5.2: server should verify users to let them check in a specific activity
- SR5.3: server provides APIs to respond front-end requests

With numbered system requirements above, the mapping between user requirements and system requirements can be represented as below.

- UR1 → SR1.1, SR1.2, SR1.3, SR1.4, SR1.5
- UR2 → SR2.1, SR2.2
- UR3 → SR3.1, SR3.2
- UR4 → SR4.1, SR4.2
- UR5 → SR5.1, SR5.2, SR5.3

1. Functional Requirements

The section of the SRS specifies the functional requirements of the AttendU system in terms of use cases and their associated use case paths. The use case model is primarily organized in terms of the externals that benefit from the use cases.

1.1.Homepage

Navigation bar

Navigation bar is located on the top of the homepage. Home menu is used for navigating to homepage, about menu link to the page which will introduce the system in detail. Contact menu links to the page which will show the contact information about website.

User can also search certain room or activity by ID on the Navigation bar.

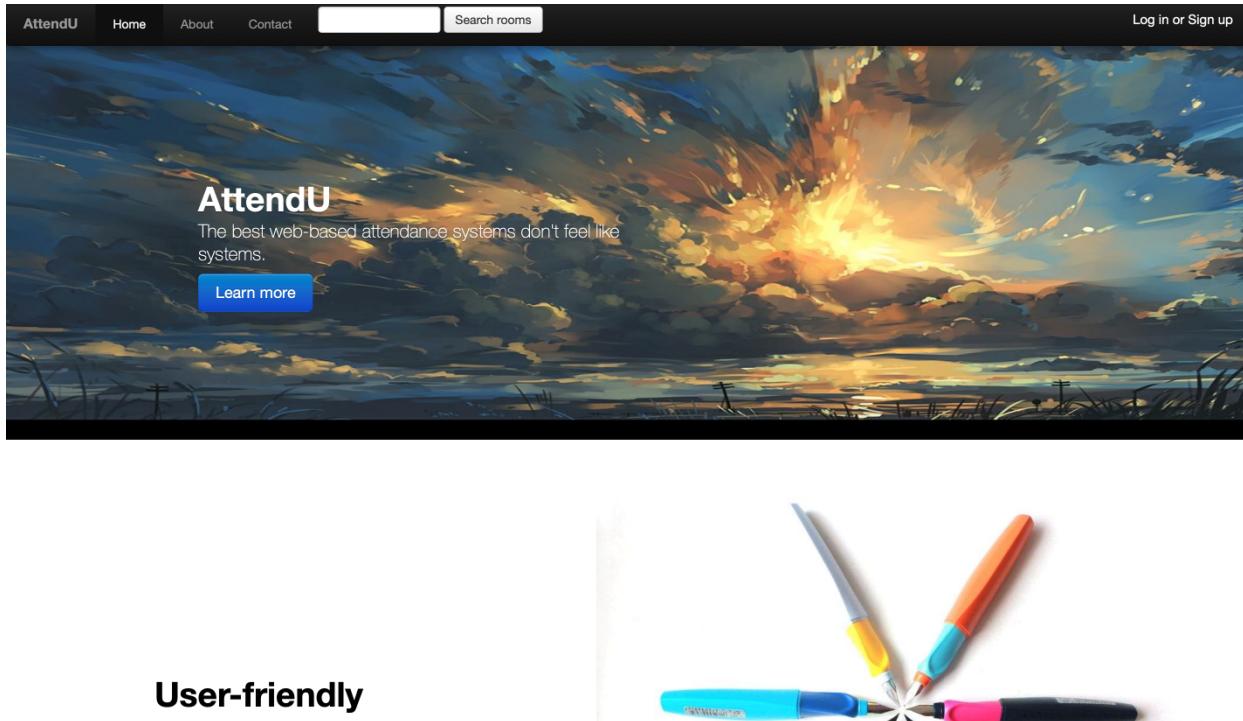


Figure 3 homepage

Display column

Display column is used for displaying the major characteristic of the system. This part is mainly used for introducing basic system function and characteristic of the system to customer. Some of the abstract introductory text located in each section. There has a link under the introductory text, when user click on the link it will jump to a new page which will be introduce certain content in detail.

Generic Usage

Most of the attendance checking apps are developed for specific use, but this app is designed for more generalized purpose. It is automated and web-based, which can be used in mobile, tablets and different operating systems. It is a dynamic and flexible system with strong generality, which can be applied to many different scenarios.

[Learn more >](#)



Figure 4 display column

1.2.User Login

User must be able to login on the homepage. The link for user login is located on the right top of the homepage.

When user click the Login or Sign up link, the page will jump to the login page.

In login page, user need to input username and password to login to the system. When the login finished, user can use the all kinds of functions of AttendU system.



User Login

Username

Password

Don't have an account? [Register](#)

Figure 5 login page

1.3.User Register

Register link located in the login page. When user do not have an account to login, user can register a new account. In register page, user need to input first name, middle name, last name, password, Email address and Cell-phone number to finish register. When user click Register button, the back-end system will check if the user name or email address has already been taken, if not, then user register successfully, otherwise there would be a prompt box to tell user that the username or email address has already been taken, user need to choose another name or email address to finish register, the register process failed.



User Register

First name <input type="text"/>	Middle name <input type="text"/>
Last name <input type="text"/>	Username <input type="text"/>
Password <input type="password"/>	Mobile phone number <input type="text"/>
Email <input type="text"/>	

Figure 6 Registration page

1.4.Dashboard

When successfully login, user will enter dashboard page. Dashboard is the core function of the whole system, through using dashboard user can perform a series of operation including check data-distribution, search room, create room, set room, and delete room.

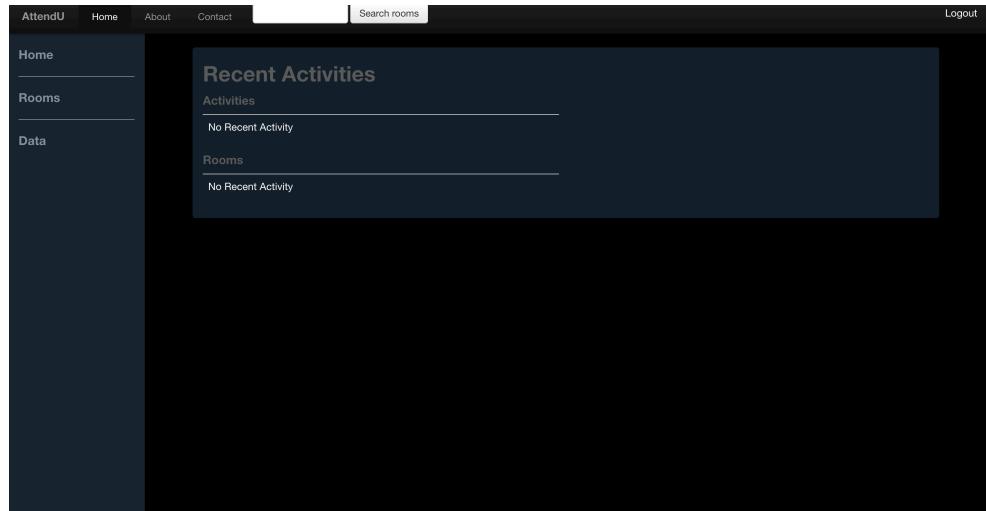


Figure 7 Dashboard page

Room List

Users can see all the rooms in this webpage after checking in.

Room ID	Room name	Number of participates	Operation
2	674	19	
3	681	25	
4	test	100	
19	test2	5	

Figure 8 Room List page

Create room

Create room menu is used to create room, at this time user would be the event sponsor, user can set series of content for the room, including room name, room introduction, the capacity of the room, the class of the activity that the room hold, and some extra information such as the connect information about the event sponsor.

The screenshot shows a dark-themed web application interface. On the left, there is a vertical sidebar with navigation links: 'AttendU' (selected), 'Home', 'About', 'Contact', 'Search rooms', 'Logout', 'Home', 'Rooms', and 'Data'. The main content area has a title 'Create Room'. It contains three input fields: 'Room name' (empty), 'Participation Number' (empty), and 'Room description' (empty). At the bottom of the form are two buttons: 'Complete' (blue) and 'Cancel'.

Figure 8 Create Room

Configure Room

If user want to modify some of the contents of the room, user can click the ‘config’ button to config room.

The screenshot shows the same dark-themed web application interface as Figure 8. The sidebar and 'Create Room' form are identical. However, in the 'Configure Room' section, the 'Room name' field contains the value 'demo' and the 'Participation number' field contains the value '100'. The 'Room description' field is empty. The 'Complete' and 'Cancel' buttons are at the bottom.

Figure 9 Configure Room

Delete room

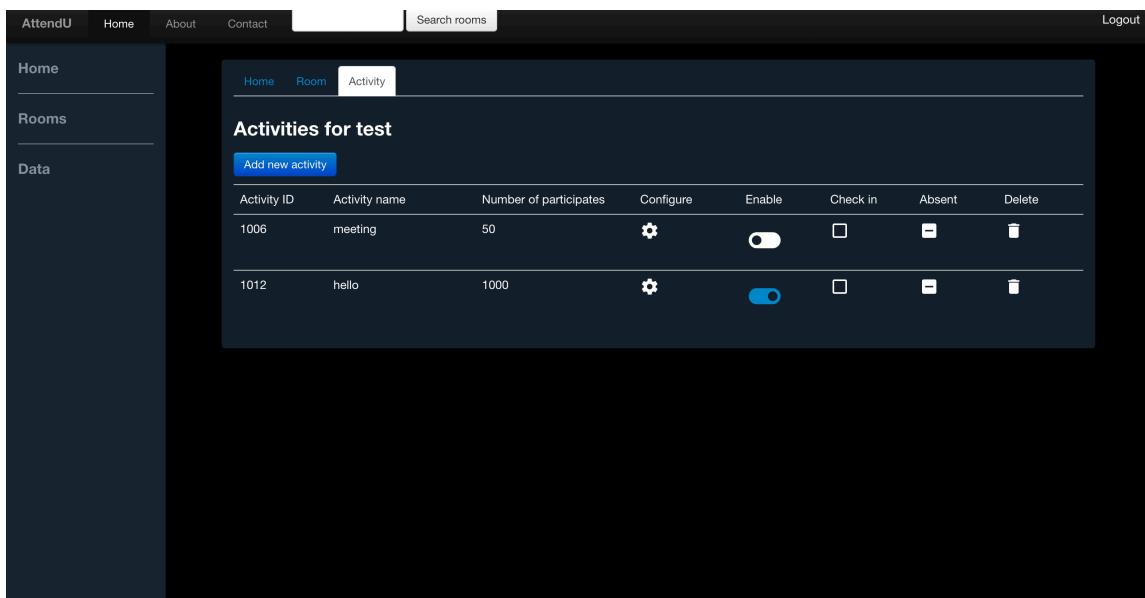
User can delete the room he/she created before, also if the user mis-delete the room he/she can also recovery the room through the “recovery” button located on the right bottom of the page. All the information about the deleted rooms can be checked through clicking the “check deleted room” button, here user can view all the information about the room created before.

Search room and Check in

Search room menu is used to search certain room, when user need to check in on certain activity he/she can input the room number to get into the room. After getting into the room, user can check in on certain activity. There are two kinds of activities: activity need user to login and login-free activity. For the activity need user to login, user must first login to the system then he/she can check-in on certain activity. For login free activity, user doesn't need to login, just search the room number then the user can finish check-in.

Activity List

A user can click on one of the room in Room List to go to the Activity List webpage.



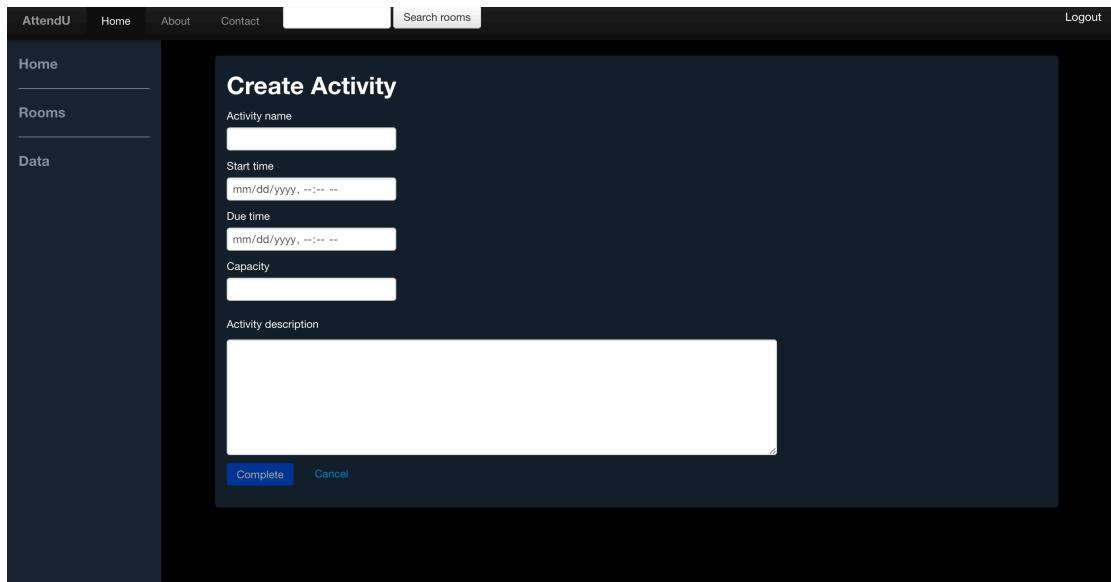
The screenshot shows a web-based application interface for managing activities. The top navigation bar includes links for 'AttendU', 'Home', 'About', 'Contact', a search bar labeled 'Search rooms', and a 'Logout' link. On the left, a sidebar lists 'Home', 'Rooms', and 'Data'. The main content area is titled 'Activities for test' and contains a table with two rows of activity data:

Activity ID	Activity name	Number of participates	Configure	Enable	Check in	Absent	Delete
1006	meeting	50			<input type="checkbox"/>		
1012	hello	1000			<input type="checkbox"/>		

Figure 10 Activity List page

Create Activity

A client can add a new activity by clicking on ‘Add new activity’ button.

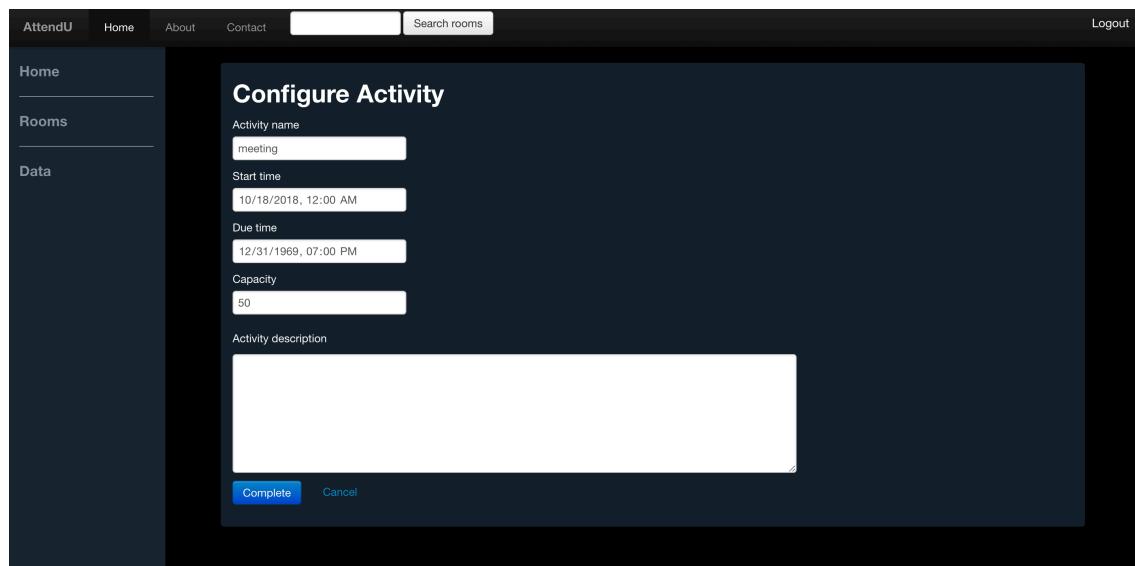


The screenshot shows a dark-themed web application interface. On the left is a sidebar with links for 'Home', 'Rooms', and 'Data'. The main content area has a title 'Create Activity'. It contains fields for 'Activity name' (an empty input field), 'Start time' (an empty input field with placeholder 'mm/dd/yyyy, --:-- --'), 'Due time' (an empty input field with placeholder 'mm/dd/yyyy, --:-- --'), 'Capacity' (an empty input field), and 'Activity description' (a large empty text area). At the bottom are two buttons: 'Complete' (blue) and 'Cancel' (white).

Figure 11 Create Activity

Configure Activity

The other is configuring activity setting by clicking on ‘gear’ button.



The screenshot shows the same dark-themed web application interface. The sidebar links are 'Home', 'Rooms', and 'Data'. The main content area has a title 'Configure Activity'. It contains fields for 'Activity name' (containing 'meeting'), 'Start time' (containing '10/18/2018, 12:00 AM'), 'Due time' (containing '12/31/1969, 07:00 PM'), 'Capacity' (containing '50'), and 'Activity description' (a large empty text area). At the bottom are two buttons: 'Complete' (blue) and 'Cancel' (white).

Figure 12 Configure Activity

Data

Data menu is mainly used for check check-in situation, the check-in situation would be shown through all kinds of charts such as bar graph, pie chart, and diagram. These charts can show the check-in status of an activity in a more intuitively way and providing a user-friendly way to help user analyst status.



Figure 13 Visualized Data

1.5.Check In

After getting into the room, user can check in on certain activity. There would be a list of activities shown on the check-in page, user can select the activity he/she wants to check in. When user select certain activity and clicks the activity link, user will enter a new page, which shows the detail of this activity, those details are the content when create the room that the user need to input. Basically, there are two kinds of activities: activity need user to login and login free activity. For the activity need user to login, user must first login to the system then he/she can check-in on certain activity. For login free activity, user doesn't need to login, just search the room number then the user can finish check-in.

2. Non-Functional Requirements

2.1. Performance requirements

Response time

- In 95% of cases, the response time of the general period is not more than 1.5 seconds, and the peak period does not exceed 4 seconds.
- Enter dashboard should not be more than 300 milliseconds from the click to the first interface.
- The time from user click login button to finish login should no more than 1s.
- The time from user click register button to finish register should no more than 1s.
- During the non-peak hours, search results can be obtained within 3 seconds according to the room number.

Business volume

- It is estimated that the number of users is 10 thousand, and the number of users per day is about 3000, and the bandwidth of the network is 100M bandwidth.
- The system can satisfy 8000 user requests at the same time and provide browsing functions for 15000 concurrent users.

System volume

- Support 150 thousand users, support GB level data.
- The number of rows in the database table does not exceed 0.5 million, the maximum capacity of the database does not exceed 500GB, and the disk space needs at least 20G.

Resource utilization rate

- The occupancy rate of CPU is <=10%.
- The occupancy rate of Memory is <=10%.

2.2. Security requirements

- Strictly control access rights, after user authentication, the user can only access and manipulate the data within the authority
- Different users have different identities and privileges. It is necessary to provide trusted authorization management services under the premise that the user's identity is authentic and trustworthy, to protect the data from illegal/over-authorized access and tampering, and to ensure the confidentiality and integrity of the data.
- Can withstand general malicious attacks from the Internet.
- Provide system log function to view system usage.

2.3. Reliability requirements

- When user input data, data should be checked by the system to prevent data exception.
- The probability of failure to complete the business due to the failure of the software system is less than 5 per thousand.
- The system defect rate takes up 1 failures every 1000 hours.

- It should be able to check all kinds of abnormal conditions during the operation of the system in time.

2.4.Compatibility requirements

- System should be able to support IOS, Android, Windows, Linux and Mac OS.
- System should support multiple platforms.

2.5.Usability requirements

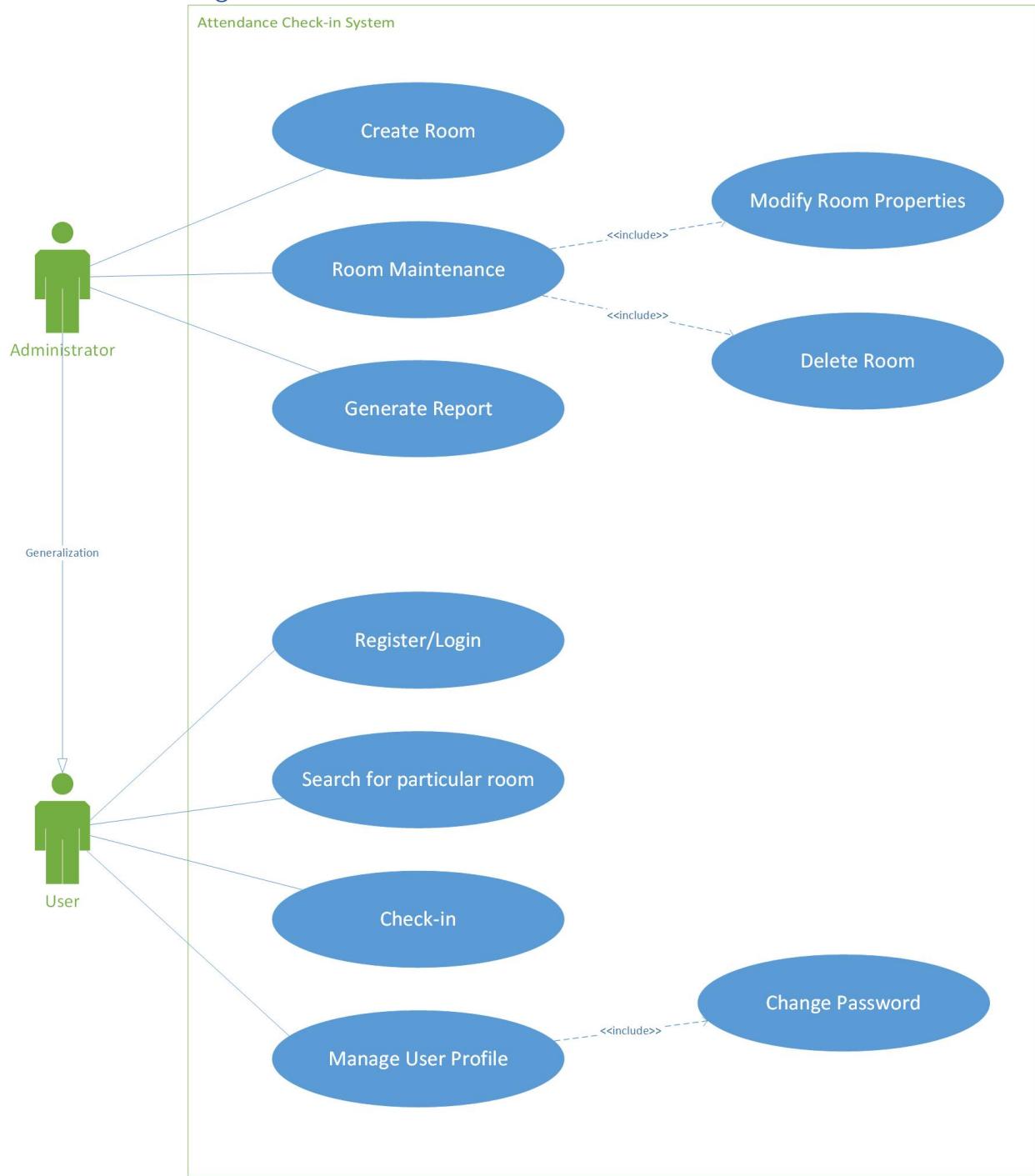
- Under the condition of poor network environment, it is guaranteed to use the basic functions of the system.

2.6.Maintainability requirements

- If the system needs to modify some functions, it should be completed within 2 days.
- Any bug's amendment time should not exceed 2 days.
- Any method of any object will not allow more than 200 lines of code.

VI. System Models

1. Use case Diagram

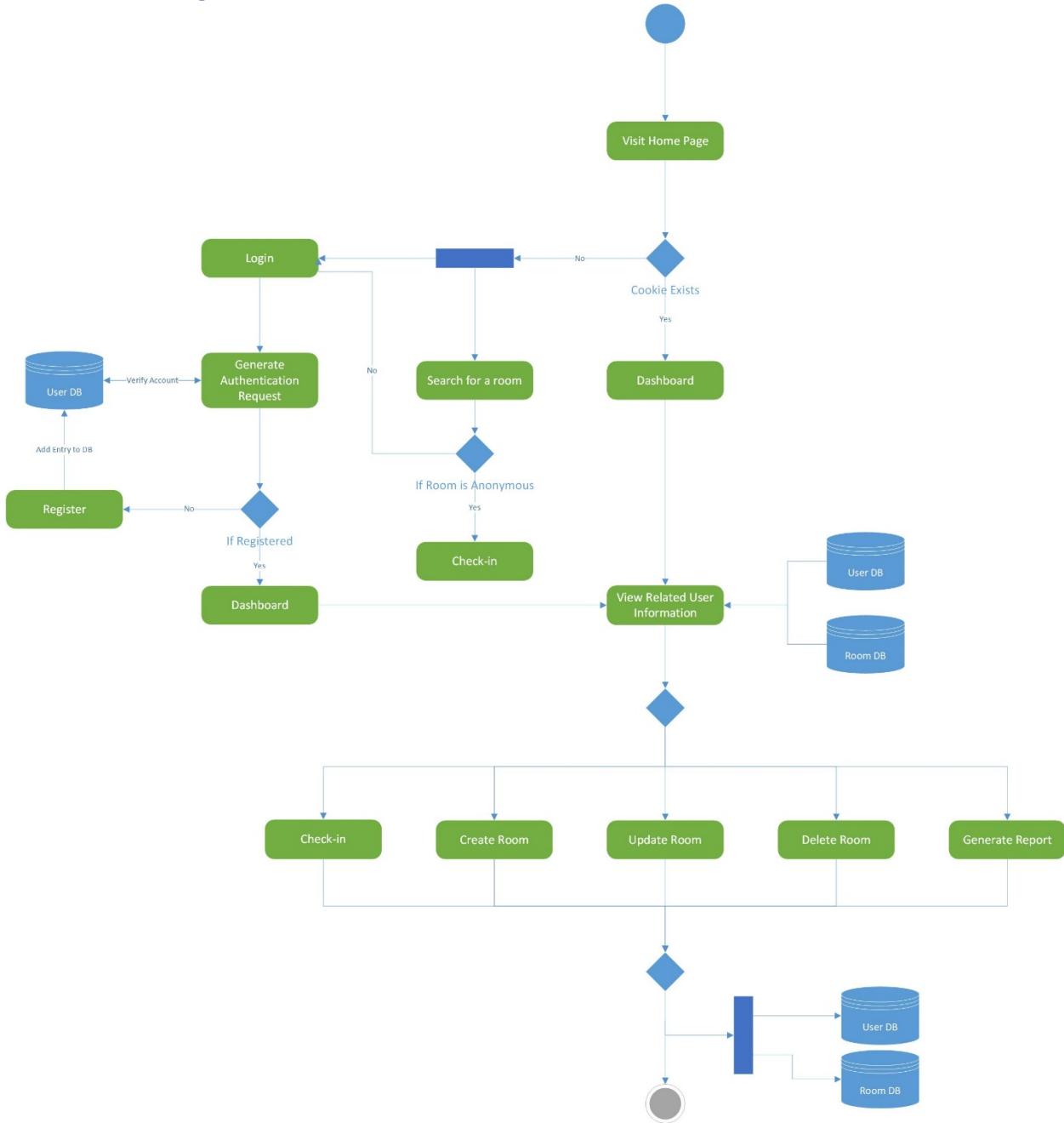


Model 1 Use case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

- A user can be a normal user trying to attend some activities by entering related room(s) created by other users.
- Users should be allowed to view and manage their own profile, like changing password, perform check-in when the room does not require real-name authentication and view their own attendance information.
- If the activities are set to non-anonymous, the user will be required to register/login to the AttendU system to successfully check-in to these activities.
- Meanwhile, a user can also be an admin user when he/she create their own room for their activities.
- If the room is owned by this user, then the user now becomes the administrator of this room and can manage the room(s), view attendance information of their rooms as well as generate attendance reports.

2. Activity Diagram



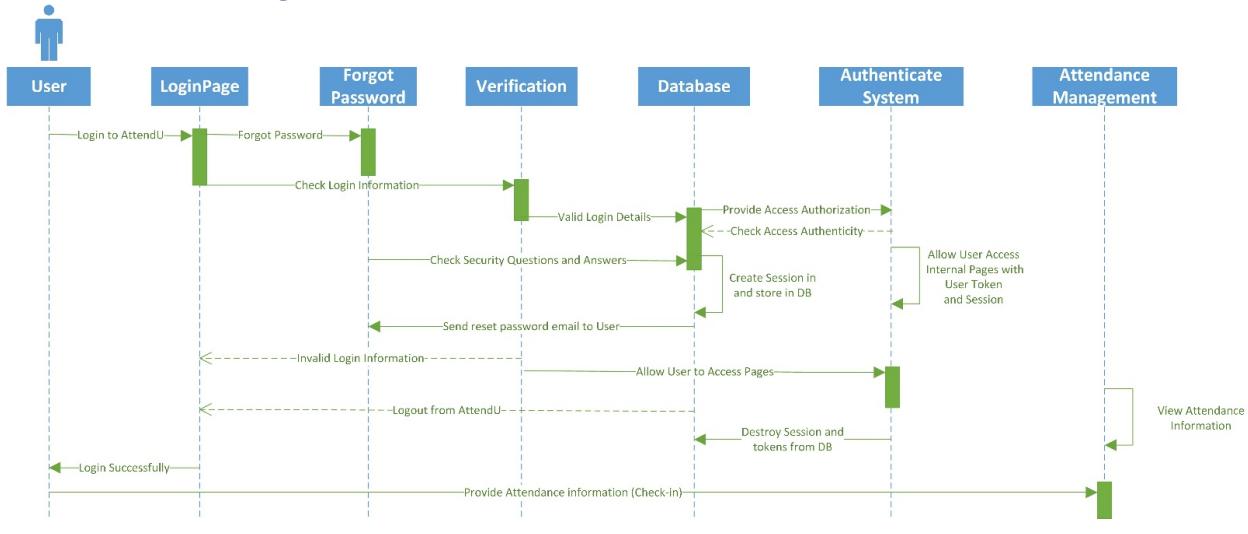
Model 2 Activity Diagram

The activity diagram explains the activity flow of users.

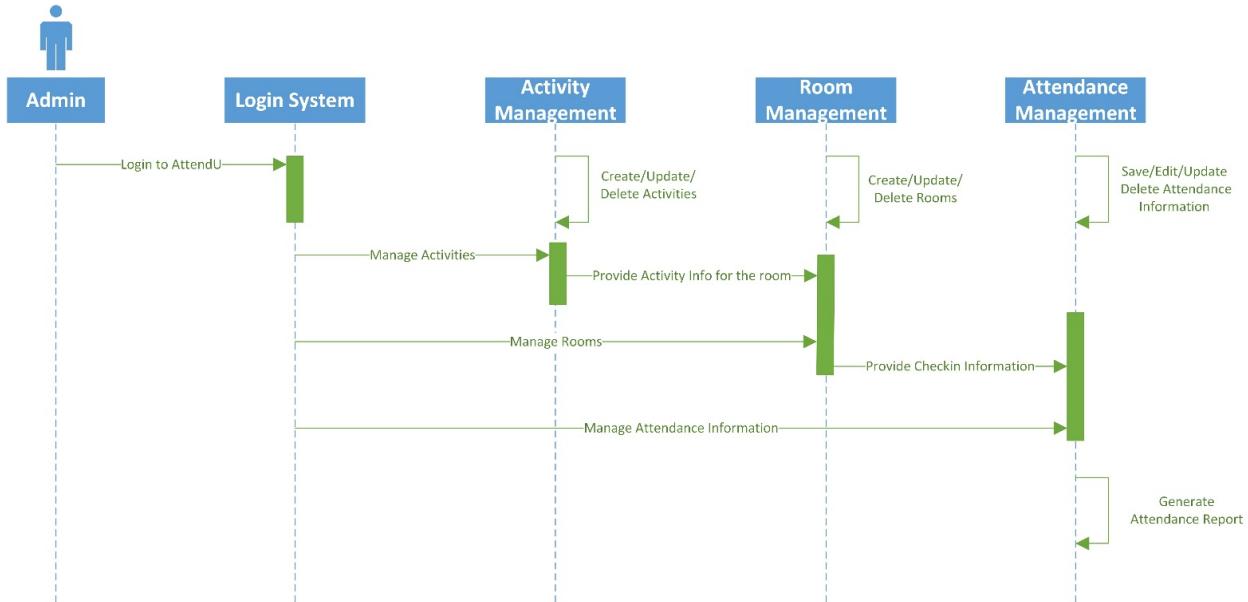
- Initially, user visit our home page, if a cookie exists, the user will have no need to login and will be redirected to the user dashboard.
- If no cookie is detected, the user can choose either login or directly search for a room by entering the unique room number.

- If the room is set to anonymous, user can click the check-in button and finish check-in without login to the system. But if it's non-anonymous, the user will be redirected to the login page and perform login or register operation.
- After login to the system, the user can visit dashboard to see their profile, rooms belong to him/her and historical attendance information.
- Users will be allowed to create/update/delete their own rooms and activities as well as generate reports of their attendance information.

3. Sequence Diagram



Model 3 Sequence Diagram - User



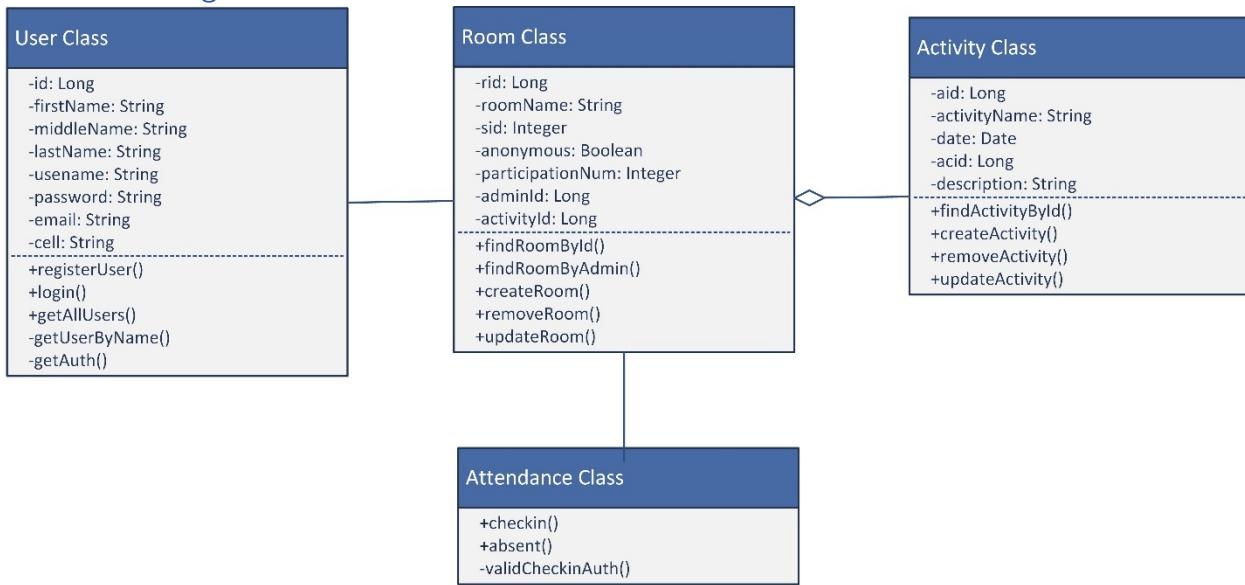
Model 4 Sequence Diagram - Admin

Login sequence describes the sequence of events in the user login system.

- When a user trying to login to the system, they might forget their passwords, then the system will check if the user can correctly answer their security questions. If yes, an email will be sent to their mailbox for resetting password.
- After the user enter login information, the system will verify the information by accessing database and provide the access authorization, if the information is invalid, return an error message.

- When the authenticity check finishes, user will be allowed to access pages and session will be created and store in database, allowing user to access internal pages with user token and session.
- The user can then check in to the activities that has real-name authentication requirement and view their own historical attendance information.
- As an administrator of room(s), the users must login before they can manage this information.
- After Login, the admin can create/update/delete activities and the corresponding rooms.
- Admin can also view/edit/update/delete the check-in information of their own rooms as well as generate attendance reports.

4. Class Diagram



Model 5 Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

- The AttendU system mainly consists of the 4 classes shown above, allows the user to be both normal user and administrator of rooms and activities.
- Detailed information of User/Room/Activity are attributes to their corresponding classes.
- The attendance class is created for collecting check-in and absence information.

VII. References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.