كلية الحاسبات والذكاء الإصطناعي
Faculty of Computers & Artificial Intelligence

جامعة حلوان
Helwan University

Acknowledgment:

In the first place, we would like to take this opportunity to express our appreciation to our supervisor Dr. Helal Ahmed for his support, guidance and encouragement throughout our graduation project for his consistent support, assistance, help, constant enthusiasm and encouragement. As he was always there for us providing more than the needed time and effort from the start of our project till the very end. He treated us as a friend and aided us in so many ways to ensure that all of our tasks were completed successfully.

Furthermore, we would like to express our gratitude to our family for the love, support, and constant encouragement and patience over the years, they are the main reason to be motivated and pushed towards success.

Finally, we would also like to extend our thanks to our faculty for providing the suitable environment that leaded us to represent the best image that computer science and Artificial intelligence graduates of Helwan University are meant to represent.

Abstract:

Hawdag is a website aims to act as an intermediate between the owners of the halls and the clients in the easiest way possible by inquiring about the halls, knowing the services they contain and the features that exist in them, and booking them easily.

1)problem: A lot of people don't know that there an online booking for hall in the internet…so they decide to go and search for a hall in different place and that take a lot of time and effort and money.

2) Objectives: Make it easier to book a wedding hall without all the time to find a suitable one…establishes a direct contact between the hall owners and clients. Provides a large library of available wedding halls.

3) Methodology: Make it easier to book a wedding hall without all the time to find a suitable one. establishes a direct contact between the hall owners and clients. Provide a GPS feature to help find the hall.

المقدمة

موقع هودج هو موقع إلكتروني يهدف إلى العمل كوسيط بين أصحاب القاعات والعملاء بأسهل طريقة ممكنة من خلال الاستفسار عن القاعات ومعرفة الخدمات التي تحتويها والمميزات الموجودة بها وحجزها بسهولة.

*المشكلة: الكثير من الناس لا يعرفون أن هناك حجزًا عبر الإنترنت لقاعة في الإنترنت ... لذلك قرروا الذهاب والبحث عن قاعة في مكان مختلف وهذا يستغرق الكثير من الوقت والجهد والمال

اتصال *الأهداف: تسهيل حجز قاعة أفراح دون الحاجة إلى إيجاد قاعة مناسبة في جميع الأوقات.. إقامة مباشر بين أصحاب القاعة والعملاء.. توفر مكتبة كبيرة من قاعات الأفراح المتاحة

*الامكانية: اجعل حجز قاعة الزفاف أكثر سهولة دون أن تجد القاعة المناسبة طوال الوقت. إقامة اتصال مباشر بين أصحاب القاعة والعملاء... يوفر الموقع الأمان والخصوصية لجميع المستخدمين.

List of Abbreviations:

RSVP: the card of the invitation.

DIY website builders: the people who control the website.

Glossary:

• Available - Ensuring the hall is available on the wedding date and any days needed for setup or rehearsal. Check for any other events already scheduled.

• Navigable - The space should be easy to get into, navigate between rooms.

• Reservable - There should be a simple process to reserve the hall.

•Spacious - The hall should have ample space for all wedding needs: ceremony, cocktail hour, reception, dancing.

• Versatile - The hall may be used for multiple purposes and should accommodate different layouts

• Additional amenities - Things like chairs, tables, linens, serving equipment, flatware, glassware.

Lists of Figure:

List of Table:

Table of Content:

Chapter 1
An Introduction

Overview:

Hawdag is a website that aims to act as an intermediate between the owners of the halls and the clients in the easiest way possible by inquiring about the halls, knowing the services they contain and the features that exist in them, and booking them easily.

The website is distinguished by providing a lot of advantages:

- Saves money by acting as an alternative for advertising.
- Has a lot of people with experience in wedding planning.
- A huge library of halls for the customers to pick a suitable one.
- Saves the clients the time needed to browse all the different halls.
- Provides information on the available packages and offers.
- Provides easy access to make reservations.
- The hall owners may upload old parties as advertisement.

Problem statement:

The wedding planning process can be overwhelming and disorganized. A centralized website is needed to streamline tasks, budgets, schedules, guest lists, contracts, and more in one place. Brides spend an average of 6-12 months planning their wedding but struggle with collaboration between themselves, bridal party members, vendors, and guests. A dynamic wedding website facilitates collaboration and ensures everyone is on the same page. Newly engaged couples need a simple solution to share wedding details with their guests in a stylish, personalized way. A customized micro-wedding website replaces traditional invitations and serves as a hub for all wedding information. Wedding guests want an easy way to RSVP, get all the details they need for travel and accommodations, and stay in the loop on bachelorette parties, showers, and other events. A complete guest-focused section of each wedding website serves as a one-stop-shop for guests. Planners and couples spend too much time fielding questions from guests and not enough time on the things that really matter. A frequently asked questions page and contact form on each site reduces repetition and lets them focus on other wedding planning tasks. Every wedding is unique so wedding websites need to be customized for each couple. Templates and DIY website builders cannot replicate the personal, customized feel that is critical for such an important event. Dynamic content - The site serves different purposes for each couple so content needs to be dynamic. Inclusion of pages, widgets, forms, calendars, maps, etc. can be tailored as needed for each wedding. Timeline and budget - Interactive timeline, budget, guest list, task lists, floor plans, seating charts and more provide an easy, visual overview of key wedding information at any point during the planning year. Timelines can be set for vendors, guests, themes, tasks, etc. Budgets can be tracked against actual spends. And more - Other features like a photo upload gallery, layered maps, wish lists, place card design, escort card assignment, schedule management and more can be included as needed for each wedding website.

Objectives And Scope:

The objective of a wedding planner project is to plan and execute a couple's wedding according to their preferences and desires. This involves coordinating various aspects of the wedding, including venue selection, catering, decorations, entertainment, photography, and other details. The ultimate goal is to create a memorable and enjoyable experience for the couple and their guests while ensuring that everything runs smoothly and according to plan. A wedding planner may also be responsible for managing the budget, negotiating with vendors, and handling any unforeseen issues that may arise during the planning process or on the day of the wedding. Wedding planning is a complex and multi-faceted process that requires a lot of attention to detail and organization. A wedding planner typically works closely with the couple to determine their preferences and needs, and then develops a plan to bring their vision to life.

The specific objectives of project include:

1-Make it easier to book a wedding hall without all the time to find a suitable one.

2- establishes a direct contact between the hall owners and clients.

3- Provides a large library of available wedding halls.

4- Provides categorized easily accessible information on halls suitable for all users

5- Provide a GPS feature to help find the hall.

6- the site offers the information in multiple languages to take the wedding planning business and the halls` owners to an international level.

7- the site provides security and privacy for all users.

8- provides multiple rating options based in (most watched -most wanted- the cheapest, etc …).

Scope:

the project was initiated to help the people getting married to book a hall to get married in. This project will be applied to serve users of different Types. Users of the system: Reserves, Hall Owners, Wedding Planners .The project uses the following languages: JS, HTML, CSS and PHP. The project will help everyone book, Browse Halls in their preferred Area and of their preferred type and theme. The functionalities inside the scope of the project are determining of the specific Preference of end user and providing the Suitable Hall And price. This project will be available on web browser and is available for everyone. Feature set Determine which planning features and tools to include within the initial scope. Things like timelines, budgets, guest management, vendor portals, micro-websites, design elements, payment processing, etc. More features increase scope but also development time. Extensibility - Consider how easily additional features, integrations, themes, etc. could be added to expand scope beyond the initial launch. Some level of extensibility is good but should not compromise launch scope/quality. In general, the scope should be wide enough to offer good functionality and value but not so wide as to make the project unmanageable, complex, or compromised in quality or polish. Determining scope requires setting boundaries that balance potential, needs and constraints.

Report Organization:

In Chapter 1 an overview was written about the project as an introduction to the objectives and scope then the working methodology we followed.

In Chapter 2 we will show the website that we visited and review that website and what we concluded from website.

In Chapter 3 we will show the beginning of our analyzation for the project from the non-functional and functional requirements and then we will show the design phase that shows the diagrams as use case diagram, activity diagram and sequence diagram.

In Chapter 4 we will start showing the implementation of the project and the test cases we applied

In Chapter 5 we will show our conclusions after finishing everything and start suggesting for the teams completing after us.

Finally References.

Work Methodology:

We used Agile Methodology

Justifications of the used Approach:

Agile methodology is a software development approach that offers several benefits, including flexibility, collaboration, and iterative development. It can lead to faster delivery of working software, better communication and teamwork, and increased customer satisfaction. Agile methodology is particularly useful in projects where requirements are not well-defined or are subject to change, or in projects where speed and flexibility are essential. By emphasizing adaptability, customer input, and continuous improvement, agile methodology can help teams build high-quality software that meets customer needs and exceeds expectations.

Flexibility: Agile methodology is designed to be flexible and adaptable to changing requirements or circumstances. It allows for changes to be made during the development process, which can be particularly useful in projects where requirements are not well-defined or are likely to change.

Advantages of the used Approach:

1-Faster time-to-market: Agile methodology allows for faster development and delivery of software by breaking down the project into smaller, more manageable iterations. This allows for quicker feedback and faster delivery of working software.

2-Customer satisfaction: Agile methodology is designed to prioritize customer satisfaction by delivering working software quickly, incorporating customer feedback into the development process, and ensuring that the software meets the customer's needs and expectations. Collaboration: Agile methodology emphasizes collaboration between team members, stakeholders, and end-users. This can lead to better communication, increased team morale, and a better understanding of the needs and goals of the project.

3-Continuous improvement: Agile methodology encourages continuous improvement through regular feedback and reflection. This helps to identify and address issues or challenges early in the development process, leading to better quality software and a more efficient development process.

4-Risk mitigation: Agile methodology can help to mitigate risks by providing regular opportunities for testing and validation, allowing for issues to be identified and addressed early in the development process.

5-Transparency: Agile methodology emphasizes transparency and visibility into the development process, making it easier for stakeholders to track progress, identify issues, and provide feedback

Work Plan (Gantt chart):

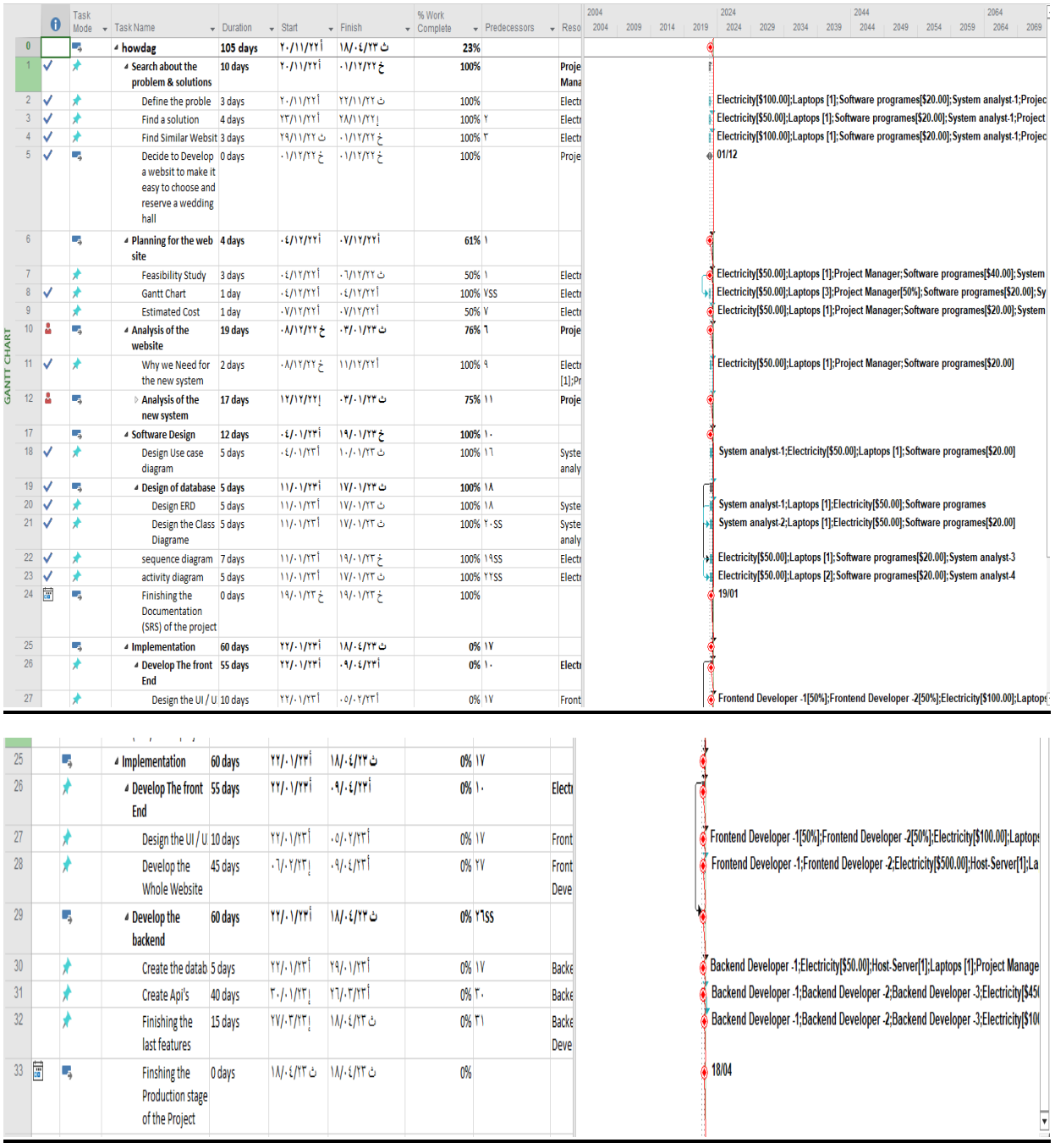


Figure 1 Gant Chart

Chapter2:
(Literature Review)

2.1 Background:

We will talk about some website like our website….and try to know what advantages is and disadvantages.

- We do not know when exactly the first wedding site appeared, but we try to define the time it appeared.

It is not possible to determine precisely who designed the first wedding website, as this dates back to the beginnings of the Internet and its history dating back to the early 1980s. It is possible that there are many websites created for this purpose by different designers and developers. However, it can be said that niche sites such as The Knot and WeddingWire were among the first to be created to provide wedding services and facilitate wedding planning. These began as online directories of service providers who cater to newlyweds, and have expanded to include tools for planning, organizing, and communicating with family and friends.

We will talk about these on the website WeddingWire and Knot.

- WeddingWire:

- WeddingWire is a website founded in 2007 that provides wedding services. WeddingWire is one of the most famous websites specializing in this field, and it provides many services that help the newlyweds plan and organize their wedding. WeddingWire provides a comprehensive database of service providers who specialize in weddings, such as stylists, beauticians, food and beverage suppliers, and more. The site also provides party planning and organization tools, including budget controls, guest tracking tools, and coordination tools with service providers. The site also offers specialized services for locating the party, sending invitations, choosing decorations and flowers, registering gifts, shooting videos and photos, music and entertainment, fire shows and fireworks, and more. WeddingWire features ratings and comments from users and previous customers of service providers, helping newlyweds make better, more informed decisions before booking. The site also provides helpful articles and tips for the newlyweds on various aspects of party planning, helping them plan the perfect wedding that meets their aspirations.

-This website has some features:

1- Comprehensive database of service providers: The site contains a comprehensive database of service providers specializing in weddings, making it easier for newlyweds to find the right suppliers to meet their needs.

2- Planning and organizing tools: The site provides party planning and organizing tools, such as budget controls, guest tracking tools, and coordination tools with service providers. These tools help the newlyweds to organize a perfect wedding that meets their aspirations.

3- User Ratings and Comments: The site allows former clients of service providers to rate and leave comments, which helps the newlyweds make better and more informed decisions before booking.

4- Useful articles and advice: The site provides useful articles and tips for newlyweds on various aspects of party planning, helping them plan a perfect wedding that meets their aspirations.

5- Save time and effort: WeddingWire facilitates the process of planning and coordination for the newlyweds, as it saves the time and effort required to find and communicate with service providers.

6- Cost Saving: The site helps the newlyweds determine the appropriate budget for the wedding ceremony and manage expenses in an effective manner, which helps them save costs and achieve the planned goals.

-7 Large and Active Community: WeddingWire has a large community of newlyweds, wedding planners, and service providers that is very active and engaged, as the site is constantly updated with new articles, tips, and tools.

8- Providing high-quality content: WeddingWire is characterized by providing high-quality and useful content for the newlyweds, as it includes useful and reliable articles and advice on various aspects of organizing weddings.

9- Customer Support: WeddingWire provides excellent customer support, as the newlyweds can communicate with customer service via e-mail or phone, and inquiries are answered quickly and effectively.

10- Offers and discounts: WeddingWire provides exclusive offers and discounts from some service providers, which helps the newlyweds save money and obtain high-quality services at a lower cost.

11-Mobile compatibility: WeddingWire is compatible with mobile devices, allowing the newlyweds to easily use the site from smartphones, tablets, and laptops.

12-Ratings and Comments: WeddingWire allows newlyweds to rate and comment on service providers, which helps them choose the most suitable and quality providers.

13-Advanced Search: WeddingWire allows newlyweds to use advanced search to find suitable service providers, where they can specify location, rating, cost, and other preferred criteria.

-Some of the disadvantage of the website:

1- Ads: WeddingWire contains many advertisements that can disrupt the user's experience and make them lose focus when using the site.

2-Sponsored Messages: Users receive a lot of sponsored messages via email and text messages, which can be a bit annoying.

3- Service Providers: Users may face some difficulties in finding service providers that are suitable for them, as there may be some service providers that do not meet the specific needs and aspirations of the newlyweds.

4-Reliance on Reviews: WeddingWire relies heavily on reviews and ratings provided by users, and the downside to this may be that some reviews may be inaccurate or fake.

5- Fees: The cost of using some of the services offered on WeddingWire may be a bit high, and this may be an obstacle for some brides who want to save money.

6-Geographical Restrictions: The couple may face some geographical restrictions in using some of the services provided on the WeddingWire website, as the services available may be restricted by geographical area.

7- Limitations in Customization: WeddingWire has a plethora of useful tools and services for newlyweds, but it lacks some customization options that may suit some users' needs.

8-Repetition of content: Some of the articles and advice on WeddingWire contain a bit of repetitive content, which may make it boring for some users.

9-Technical Support: WeddingWire offers technical support to users by email and phone, but it can be a bit slow in responding to inquiries and technical issues.

10- Unavailability of services in some areas: Some users may have difficulty finding available service providers in their area, especially if they live in remote or small areas.

11-Budget Control Constraints: Some of the tools available on WeddingWire can be a bit limited in budget control, which means that some users may find it difficult to determine what they can afford for a wedding.

-We have another website that appeared with WeddingWire Called the Knot.

- The Knot was founded in 1996 by three partners: Carla Rizzo, David Liu, and Michael Maxi. The website was created to provide a reliable source of information and tools for brides and grooms to plan and organize their weddings. Thanks to the efforts of the founders, the site has quickly grown to become one of the most popular and largest wedding planning websites in the world. In 2019, The Knot was acquired by WeddingWire, and the site became part of the larger WeddingWire group. The Knot is a popular website for weddings and special events. Founded in 1996, the site offers services and tools to brides and grooms to help them plan and organize their wedding. The services provided by the site include searching for party locations, ideas for decoration and costumes, photography, music, and other services related to organizing weddings.

The Knot is a reliable source of information, advice and useful tools for planning and organizing weddings. It also provides a large database of different service providers in the field of wedding planning, who have been carefully selected to ensure the quality of the services they provide. In general, The Knot is an important destination for brides and grooms who want to plan and organize unique and successful weddings.

-We will talk about some features:

1- Service Providers Database: The site provides a large database of different service providers in the field of organizing weddings, who have been carefully selected to ensure the quality of the services they provide.

2-Planning Tools: The site provides a wide range of useful wedding planning and organizing tools, including planning timelines, detailed to-do lists, and tools for budgeting and controlling costs.

3-Ideas and tips: The site contains a large collection of useful ideas and tips for organizing weddings, including tips for choosing among service providers, ideas for decoration and decorations, photography, and more.

4- User community: An important part of The Knot is the user community, where brides and grooms can get advice and tips by participating in forums and discussion groups.

5-Mobile App: The Knot has a mobile app that can be downloaded to smartphones, which provides the same services and tools available on the website, making planning and organizing easier and more convenient.

-we will talk about some disadvantage:

1- Limited Information Availability in Some Areas: Availability of detailed information about party service providers and websites can be limited in some areas or venues, and this can make it difficult for brides and grooms to find suitable suppliers for them.

2-Cost: Some of the services and tools available on The Knot can be quite expensive, and this can be a barrier for brides and grooms who are working on a tight budget.

3-Difference in quality of services: The quality of different service providers listed on the website may vary, meaning that brides and grooms may need to research carefully to find providers that offer high quality services.

4- Ads: The Knot site includes advertisements for various providers and companies, and this may display some information and tools on the site for advertising effect.

5-Geographical area restrictions: Some of the services and tools available on The Knot website may be limited in certain geographical areas, and this may affect the ability of brides and grooms to benefit from some services.

6- Focus on traditional styles: The Knot may focus primarily on traditional wedding styles, and this can limit the variety of ideas and options available to brides and grooms.

7-Dependence on Internet Technology: The Knot relies heavily on Internet technology, and this can make it difficult for brides and grooms who do not have access to the Internet to take full advantage of the services available on the site.

8-Scheduling: Some of the information and tools available on the Site may not be updated on a regular basis, and this means that some of the plans and schedules available on the Site may not be entirely accurate.

-There is some difference between two website:

1- Design and Experience: The Knot has a sleek and modern design, while WeddingWire focuses on a more minimalist and functional design. The Knot also provides an interactive user experience and access to ideas, shopping and registration in one place.

2-Content and Resources: The Knot provides rich content and useful resources for wedding planning, including expert articles, tools, and information about the wedding industry, while WeddingWire primarily focuses on providing detailed information on different providers and customer reviews.

3-Vendors and Services: The range of providers and services available on each of the sites varies, with The Knot focusing mainly on elegant, luxurious and modern weddings, while WeddingWire focuses on providing more diverse options for classic and traditional weddings.

4-Pricing: The pricing available on each of the sites varies and this can be affected by the different providers and services available.

5-Geographical area: The focus of each of the sites and their content and the available providers can vary according to the geographical area.

2.3 Analysis of the Related Work and Example:

-we search about the most popular website in our time .. there some of these sites:

1- The Knot: The Knot is one of the most popular wedding planning websites. The site provides a wide range of tools and services, including ideas, designs, suppliers, shopping, and registration in one place.

2-WeddingWire: WeddingWire provides a wide range of different wedding-related suppliers and services, including suppliers, services, ideas, designs, and shopping.

3-Zola: Zola offers comprehensive wedding planning services, including wedding invitations, registration, gifts, monthly tours, and more.

4-Weddingbee: Weddingbee focuses on providing helpful content and resources for brides and grooms, including ideas, designs, advice and personal experiences.

5- Martha Stewart Weddings: Martha Stewart Weddings is one of the leading wedding planning sites, providing a wide range of resources, ideas, designs, and advice.

6-Brides.com: Offering a wide range of content, ideas, designs, suppliers, and services, Brides.com is a useful resource for wedding planning.

7- WeddingLovely: WeddingLovely provides various wedding planning services, including providers, services, community, and more.

8-Junebug Weddings: Junebug Weddings offers unique and new wedding ideas and designs, with a focus on luxurious and exclusive weddings.

9-Style Me Pretty: Style Me Pretty offers a wide range of wedding ideas, designs, suppliers, services, and advice.

10-BridalGuide: BridalGuide is a one-stop source for wedding-related information, ideas, designs, and services.

11-Wedding Chicks: Wedding Chicks focuses on providing content, ideas, designs, and different suppliers, in addition to wedding planning services.

12-Rustic Wedding Chic: Rustic Wedding Chic focuses on rustic and charming weddings, and provides a range of different ideas, designs, and suppliers.

-This some of the most popular sites and small brief about each site…you will find that there no obvious difference between them. All this site offers in general the same services and do the same job… Ultimately, the most popular site for wedding planners may depend on their individual needs and preferences, as each site offers unique features and tools to help with wedding planning.

2.4 Project planning:

- 2.4.1 Feasibility Study

A feasibility study is an assessment of the practicality of a proposed project or system. It aims too objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources. required to carry through, and ultimately the prospects for success.

-2.4.2 Project Description

The Website will offer users a community where they can meet and engage with people who share the same interest, it will also help them in booking meetings with experts, will help them to have a great wedding. Our website will offer expert people that will help you with their offers, they can also book anything related to wedding, through our website as it gathers the stores selling anything related to wedding, users also can have their virtual place where they can tell their own opinion about what we offer in our website. Users can also identify the Halls in front of them and to know information about them, by only capturing or uploading a picture of the halls. We will also provide the users with articles written by our experts, where they can learn new information about our offers, halls and about us.

-2.4.3 The Problems (User's Needs)

The user faces a lot of problems in booking hell for wedding, that he

can’t find Suitable Hall or find one with suitable price of offer or one has a special location and other different problem.

-Having a website for making wedding will make have a wedding is much easier … that will save a lot of time, money, and efforts.

-A lot of people don’t know that there a site for arranging the wedding and They follow the traditional ways by searching for hells by their self.

- Some people suffer while they are choosing hells for themselves because they lack knowledge about what is the most suitable type for themselves and the lake of options that are produced.

- People who are interested in having a wedding do not have a special place that gathers all their needs and brings this community experience together.

-2.4.4 Target Audience

- We are targeting anyone interested in having a wedding.

-Stores Owners.

-Hall Owners.

-wedding Planers.

-Suppliers

-2.4.5 Idea Motivation

-The idea motivation for a website for weddings could be to provide a comprehensive online platform for couples who are planning their wedding. A wedding website could serve as a central hub for all things related to the wedding, providing information, resources, and tools for the couple and their guests.

Some potential features of a wedding website could include:

A wedding planning checklist to help the couple stay organized and on track.

A budget tracker tool to help the couple manage their wedding expenses.

A directory of vendors, including photographers, florists, caterers, and DJs, with reviews and ratings from other couples.

A calendar of events, including the wedding ceremony, reception, and

any related events such as a rehearsal dinner or bridal shower.

A guest list management tool to help the couple keep track of RSVPs and meal preferences.

A section for the couple to share their love story, photos, and other personal details with their guests.

A section for guests to RSVP, view the wedding registry, and get directions to the wedding venue.

2.5Analysis of the new system

-2.5.1 User requirements

- The User shall be able to see all halls on the website.
- The user shall be able to create a booking of a hall.
- The User shall be able to see the available time to book the hall.
- The User shall be able to have contact with the hall owner.
- The User shall be able to see all wedding planers and their plans.
- The User shall be able to create a reservation of a plan.
- The User shall be able to have contact with the wedding planner.
- The User should be able to give a rate to the hall which he booked.
- The User should be able to write a comment to the hall which he booked.
- The User should be able to like or dislike the hall which he booked.
- The User may be able to filter all halls to choose one User should be able to see all his hall bookings.
- The User should be able to see all his plans reservations.

1. Hall Owner Requirements:

- The Hall Owner shall be able to create an account on the website.
- The Hall Owner shall be able to see, accept and reject booking requests.
- Hall Owner shall be able to add a new hall to the website.
- Hall Owner should be able to update his halls.
- Hall Owner should be able to delete his hall.
- Hall Owner may be able to change the status of the hall (available or not)
- Hall Owner should be able to edit and update list of services that hall introduces.
- Hall Owner should be able to see rates and comments about his hall.
- Hall Owner shall be able to see clints of his hall.

2. Wedding Planner Requirements:

- The Wedding Planner shall be able to create an account on the website.
- The Wedding Planner shall be able to update his profile.
- The Wedding Planner shall be able to add his plans.
- The Wedding Planner shall be able to update his plans.
- The Wedding Planner shall be able to delete his plans.
- The Wedding Planner shall be to accept or reject reservation of his plans.
- The Wedding Planner should be able to update his plans on the website.

3. Admin Requirements:

- The admin shall be able to confirm hall request.
- The admin shall be able to reject hall request.
- The admin shall be able to delete users.
- The admin shall be able to delete the hall owner.
- The admin shall be able to delete the wedding planner.
- The admin shall be able to see all users.
- The admin shall be able to see all hall owners.
- The admin shall be able to see all wedding planners.
- The admin shall be able to add users, hall owner, and wedding planner.

4. supplier Requirements:

- The Supplier shall be able to create an account on the website.
- The Supplier shall be able to update his profile.
- The Supplier shall be able to add his services.
- The Supplier shall be able to update his services.
- The Supplier shall be able to delete his services.
- The Supplier shall be to accept or reject reservation of his services.
- The Supplier should be able to update his services on the website.

5.System Requirements:

- Users shall be able to see all halls and book in one.
- The Hall Owner shall be able to accept or reject the user booking.
- Hall Owner shall be able to add a new hall to website.
- Hall Owner should be able to update his halls.
- Hall Owner should be able to delete his hall.
- Hall Owner may be able to change the status of the hall (available or not)

- Hall Owner should be able to edit and update list of services that hall introduces.
- Hall Owner should be able to see rates and comments about his hall.
- Hall Owner shall be able to see clints of his hall.

6.Domain Requirements:

• User should be able to see all halls and book one
• Hall owner should be able to add his halls and
update them and add all services he wants.
• Wedding planner should be able to put his work on the website
• The website should have a suitable throughput rather than falling
• The website should be developed in a way to prevent fake booking and fake halls
• All date in website should be accurate

Non- Functional Requirements:

- Availability

  Availability requirement is any requirement that is not a functional, data or process requirement concerned with defining the periods when the solution can be used.
- Maintainability

  Maintainability requirement is how easy it is for a system to be supported, changed, enhanced, and restructured over timee.
- Security

  The application must remain resilient in the face of attacks. The behaviour of the software must be correct and predictable. The software must be available and behave reliably even under DOS attacks. It must ensure the integrity of the customer account information.
- Flexibility

  It is the ease with which the software can be modified to adapt to different environments, configurations, or user expectations.

- Scalability

  It is the ability of the application to handle an increase in workload without performance degradation, or its ability to quickly enlarge.

- Usability

  It is a non-functional requirement, because in its essence it does not specify parts of the system functionality, only how that functionality is to be perceived by the user, for instance how easy it must be to learn and how efficient it must be for carrying out user tasks.

Chapter 3 :

Software Design

Class Diagram

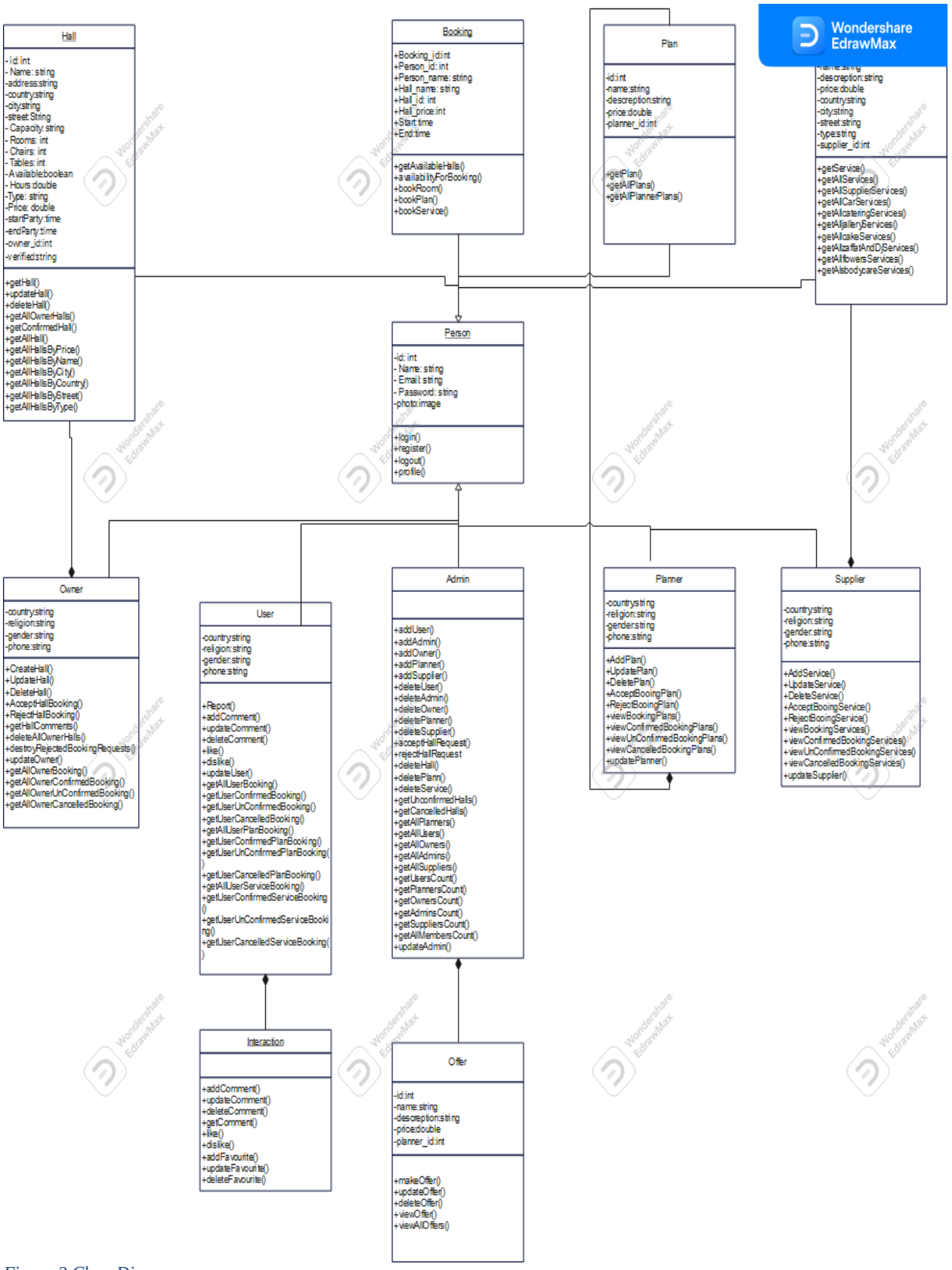


Figure 2 Class Diagram

-Use case

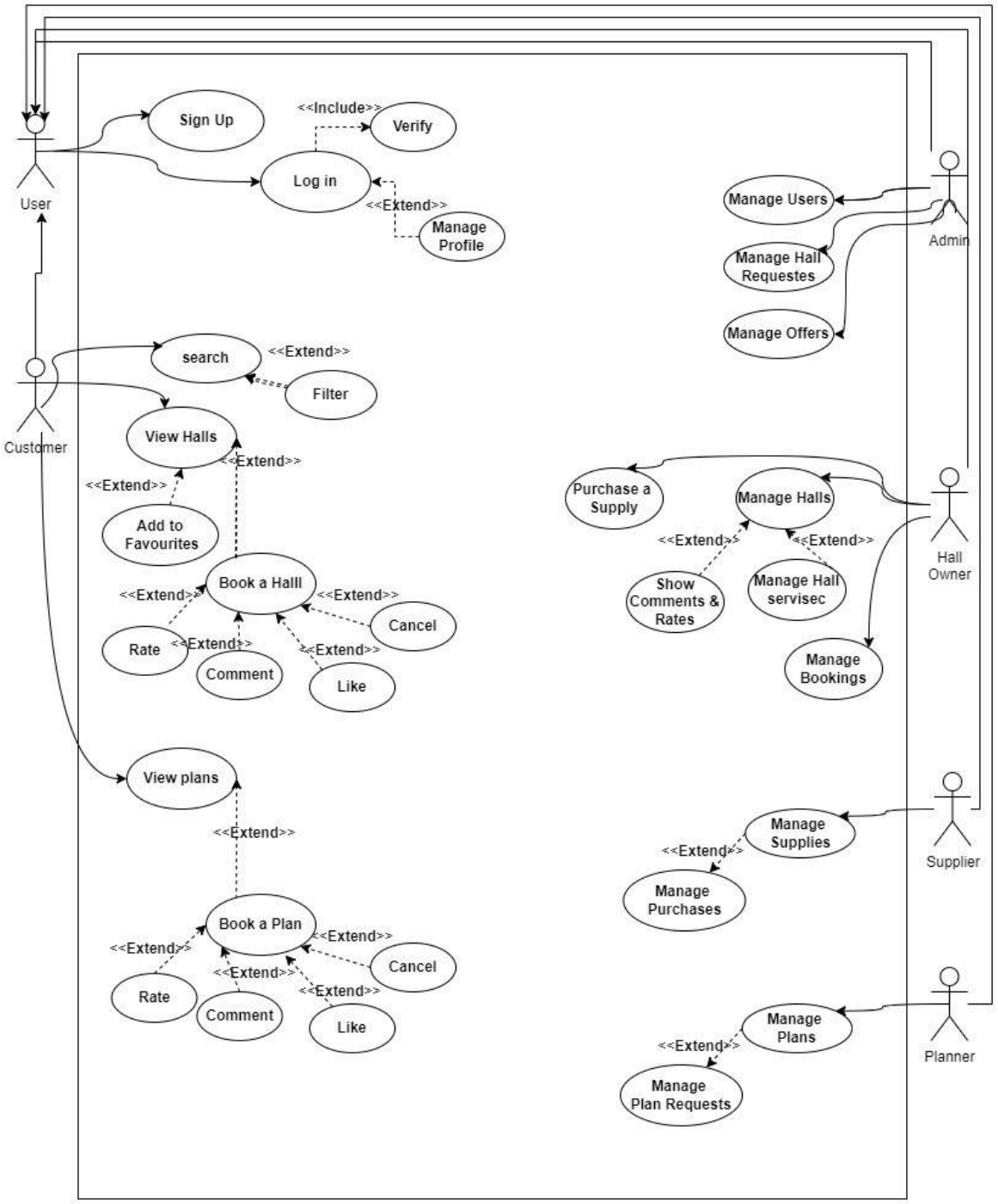


Figure 3 Use Case

Use Case Scenario:

Use Case 1	Sign up
Actor	User
Goal	User signs up to new account
Precondition 1	None
Postcondition	Verifying if user info valid or not

Use Case 1 B	Report
Actor	User
Goal	User able to report offensive

Precondition 1	1) user logs in
Postcondition	The report goes to admin to review

Use Case 2	Log in
Actor	User
goal	User signs up to new account
Precondition 1	User already have an account
Postcondition	1. The system Verify his data and the user (successful) 2. The system cannot Verify the

Use Case 3	Search
Actor	Reserve

Goal	Search to find a suitable hall
Precondition 1	User logged in
Postcondition	Viewing a list of halls
Extensions	Using special filter

Use Case 4	filter
Actor	Reserve
Goal	Filter results
Precondition 1	User logged in
Postcondition	Viewing a list of halls

Use Case 5	Display

Actor	Reserve
Goal	Display all halls
Precondition 1	User logged in
Postcondition	Viewing a list of halls
Extensions	Booking a hall

Use Case 6	booking
Actor	Reserve
Goal	Select a hall to book
Precondition 1	User logged in
Postcondition	User selects a service

Extensions	1)Chat with owner 2) rate the hall

Use Case 7	Cancelling a booking
Actor	Reserve
Goal	Cancelling a booking and pay a fine after a deadline
Precondition 1	User booked a hall
Postcondition	Paying a fine after deadline

Use Case 8	Rate and comment
Actor	Reserve
Goal	Gives feedback about your experience

Precondition 1	User booked a hall

Use Case 9	Select services
Actor	Reserve
Goal	Select services package
Precondition 1	User selected a hall

Use Case 10	Chat with the owner
Actor	Reserve
Goal	Chat with the owner

Precondition 1	User selected a hall

Use Case 11	Booking wedding planner
Actor	Reserve
Goal	Select a wedding planner to book
Precondition 1	User logged in
Postcondition	User selects a plan
Extensions	1)Chat with the planner 2) rate the planner 3) cancel booking

Use Case 12	Cancel a plan

Actor	Reserve
Goal	Cancelling a booking and pay a fine after a deadline
Precondition 1	User booked a plan
Postcondition	Paying a fine after deadline

Use Case 13	Rating a planner
Actor	Reserve
Goal	Select a wedding planner to book
Precondition 1	User selected a plan

Use Case 14	Chat with the wedding planner

Actor	Reserve
Goal	Chat with the wedding planner
Precondition 1	User selected a plan

Use Case 15	Show bookings
Actor	Hall owner
Goal	Show all current bookings for your halls
Precondition 1	User logged in
Extensions	1)accept a booking 2) reject a booking

Use Case 16	Show my halls

Actor	Hall owner
Goal	Show all halls
Precondition 1	User logged in
Extensions	1) create a hall 2) delete a hall 3) update a hall 4) select a hall

Use Case 17	Crate a hall
Actor	Hall owner
Goal	Create a new hall
Precondition 1	User logged in

Use Case 18	Delete a hall

Actor	Hall owner
Goal	Delete an existing hall
Precondition 1	Selects an existing hall

Use Case 19	Update a hall
Actor	Hall owner
Goal	Update hall info
Precondition 1	Selecting an existing hall

Use Case 20	Preview a hall
Actor	Hall owner

Goal	Show all information about this specific hall
Precondition 1	Owner selected a hall
Extensions	1) show rates and comments 2) show clients for this hall

Use Case 21	Show rates and comments
Actor	Hall owner
Goal	show rates and comments for a hall
Precondition 1	Owner selected a hall

Use Case 22	Show clients
Actor	Hall owner

Goal	Show all clients for your halls
Precondition 1	Owner selected a hall

Use Case 23	edit hall status
Actor	Hall owner
Goal	edit hall status
Precondition 1	Owner selected a hall

Use Case 24	edit hall service
Actor	Hall owner
Goal	edit hall service

Precondition 1	Owner selected a hall

Use Case 25	Add works
Actor	Wedding planner
Goal	Add previous works
Precondition 1	User logged in

Use Case 26	Update works
Actor	Wedding planner
Goal	update previous works
Precondition 1	User logged in

Use Case 27	Chat

Actor	Wedding planner
Goal	Chat with users
Precondition 1	User logged in

Use Case 28	Show Purchases
Actor	Supplier
Goal	Show all purchases for your Supplies
Precondition 1	Supplier Logged in the system

Use Case 29	Manage Supplies
Actor	Supplier

Goal	Manage Supplies and the CRUDs for it
Precondition 1	Supplier Logged in the system
Use Case 30	Manage Supplies Purchases
Actor	Supplier
Goal	Manage Purchases reject Or confirm
Precondition 1	Supplier Logged in the system

Use Case31	booking
Actor	Reserver (Hall Owner)
Goal	Select a supply to purchase

Precondition 1	User logged in
Postcondition	User selects a supply
Extensions	cancel booking

Use Case 32	Cancel a supply Purchase
Actor	Reserve (Hall Owner)
Goal	Canceling a Purchase and pay a fine after a deadline
Precondition 1	User Purchased a Supply
Postcondition	Paying a fine after deadline

Activity Diagram:

-USER REGISTER:

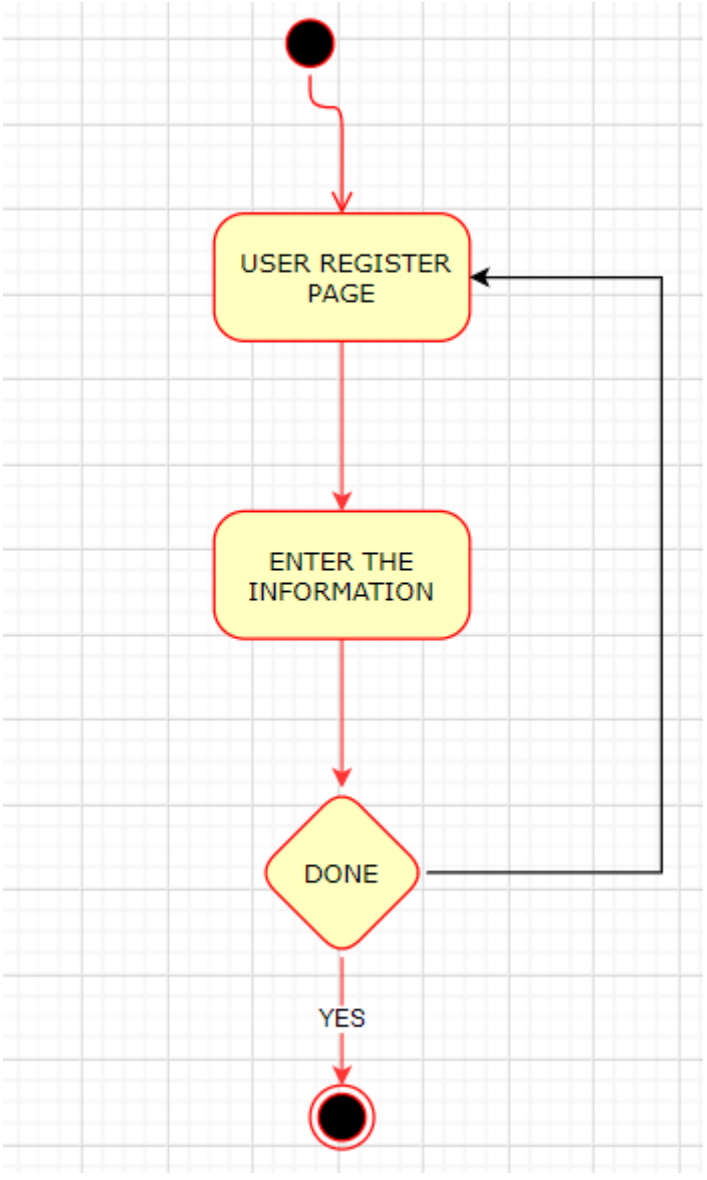


Figure 4 Activity(REGISTER)

- LOG IN:

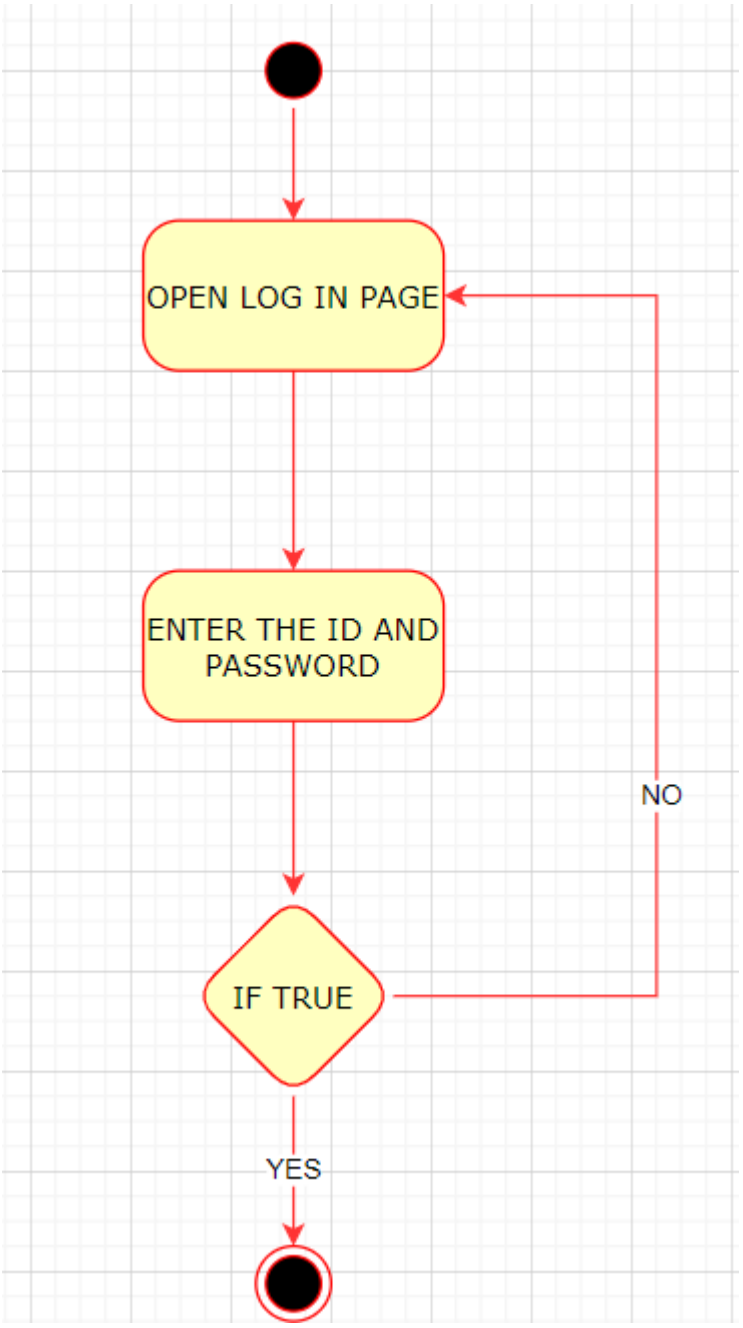


Figure 5 Activity (Login)

- HallOwner(SelectHall):

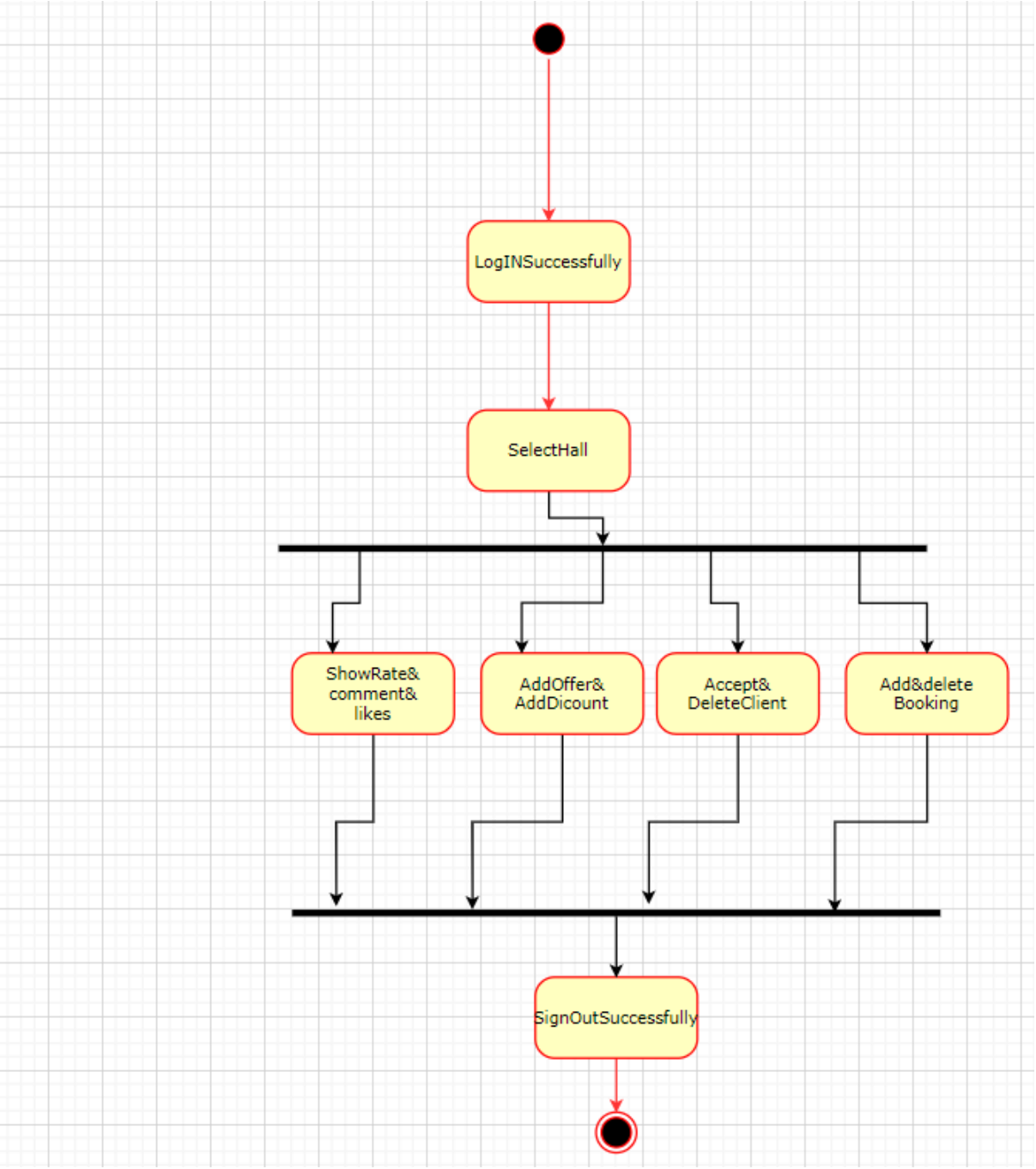


Figure 6 SelectHall

- WeddingPlanner:

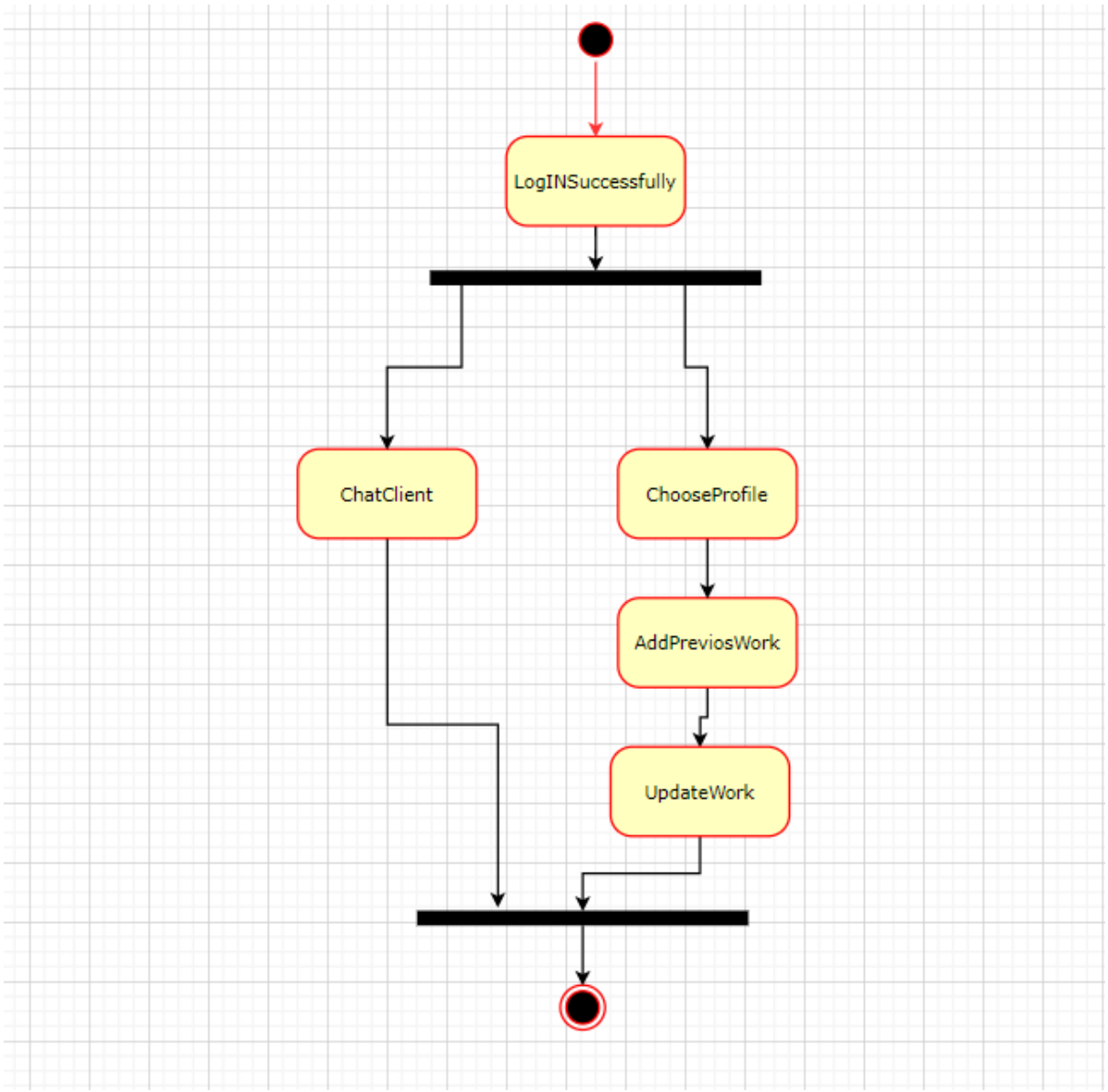


Figure 7 WeddingPlanner

- reserve (BookHall):

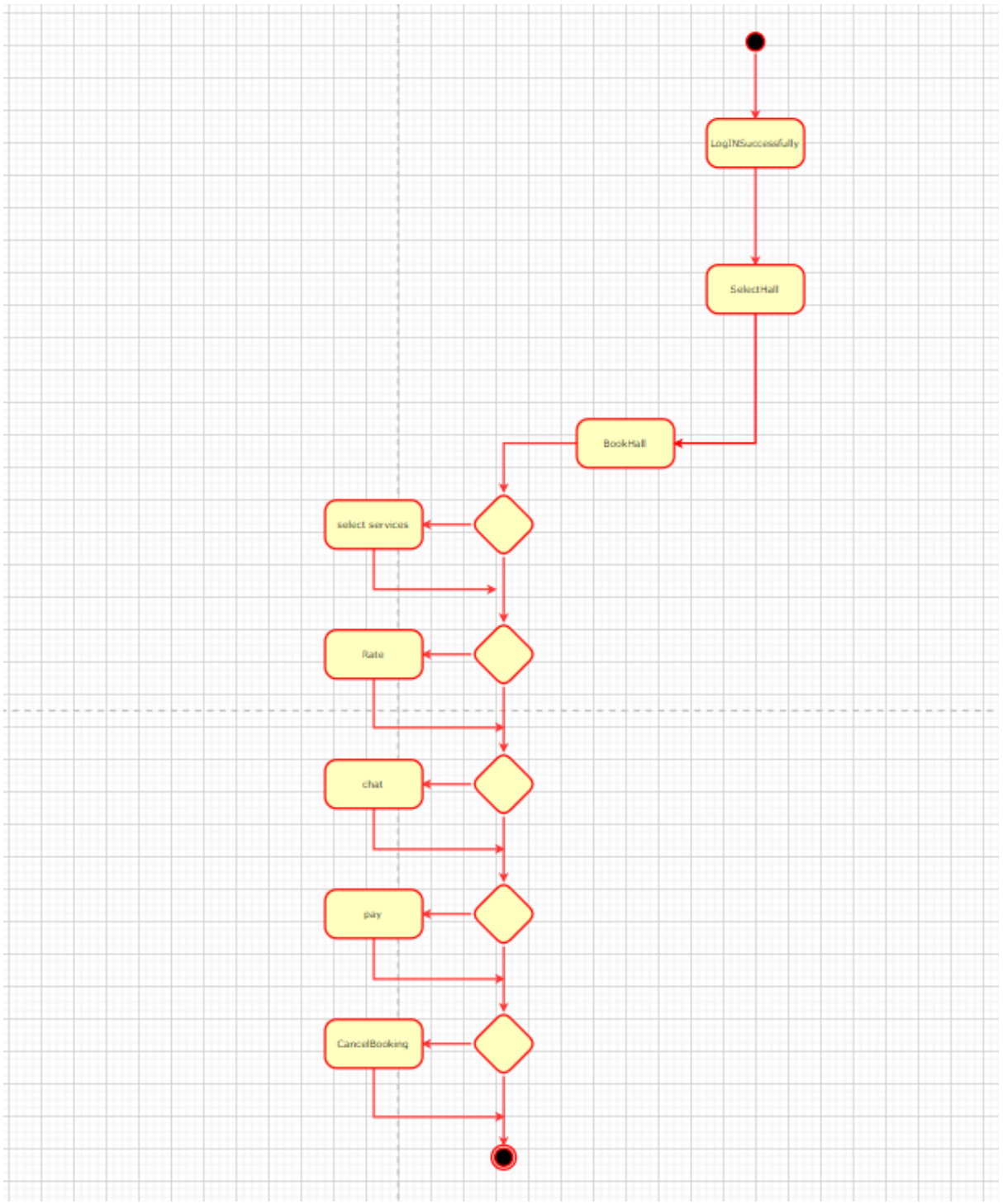


Figure 8 BookHall

- Reserve (Search):

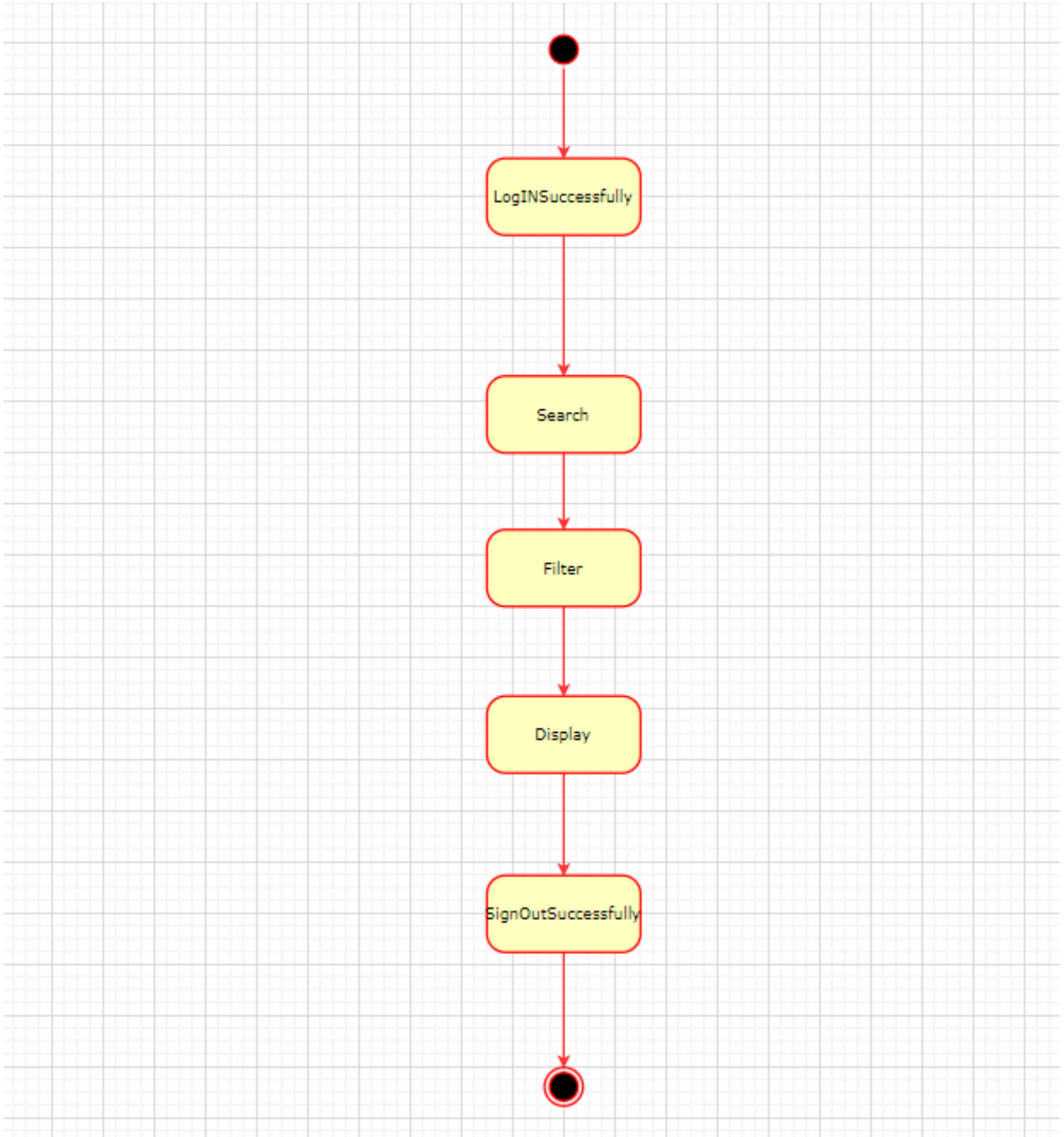


Figure 9 Search

- Reserver(BookPlanner):

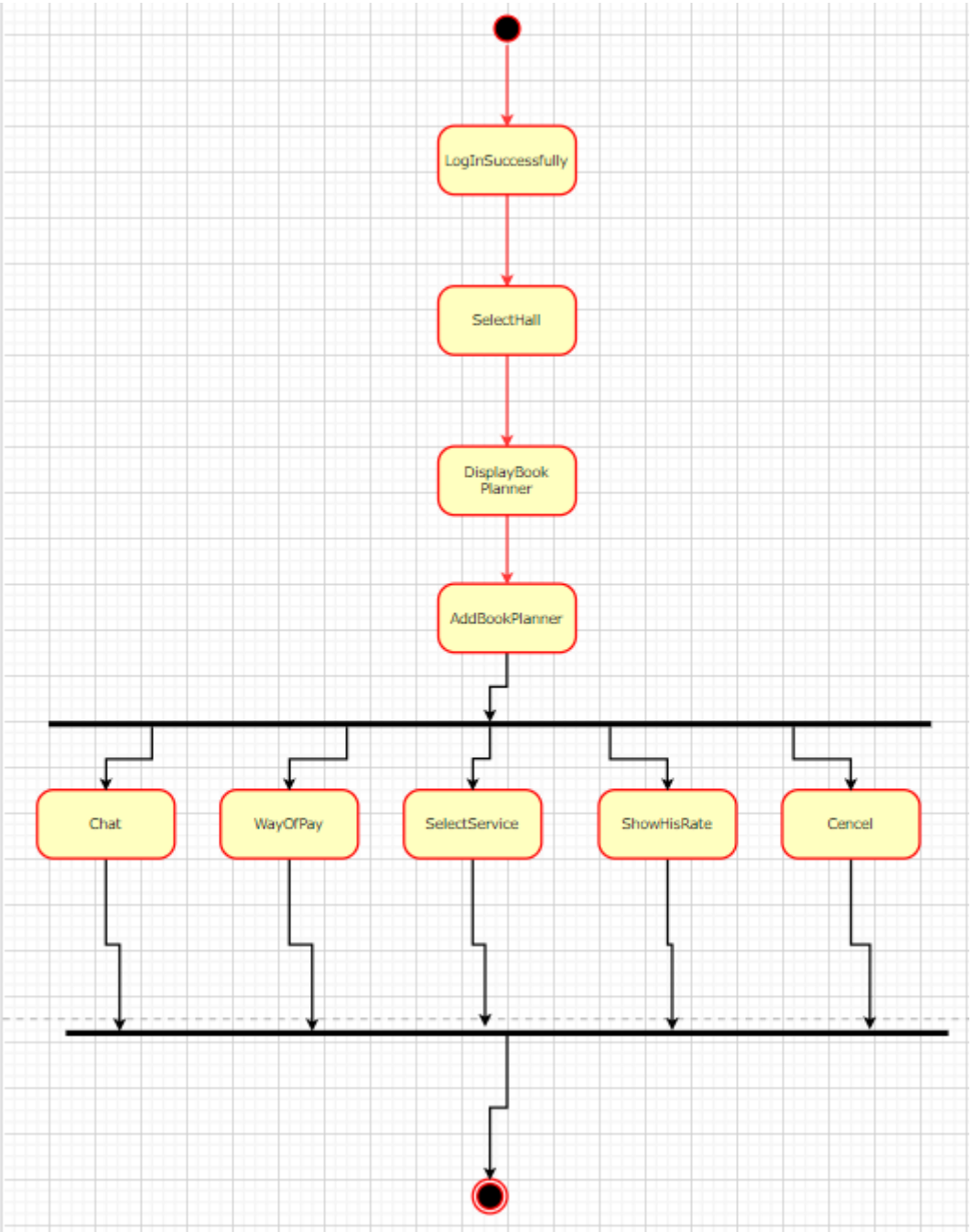


Figure 10 BookPlanner

-Log Out:

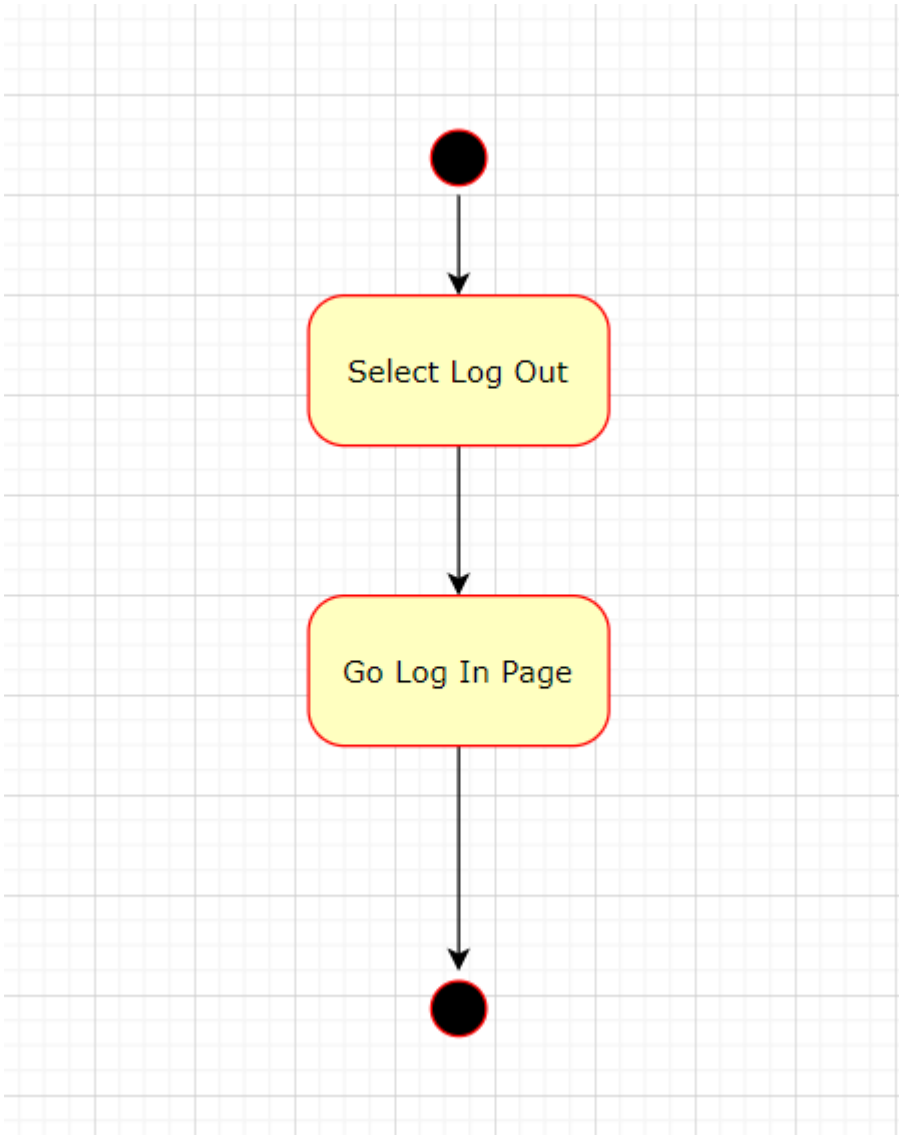
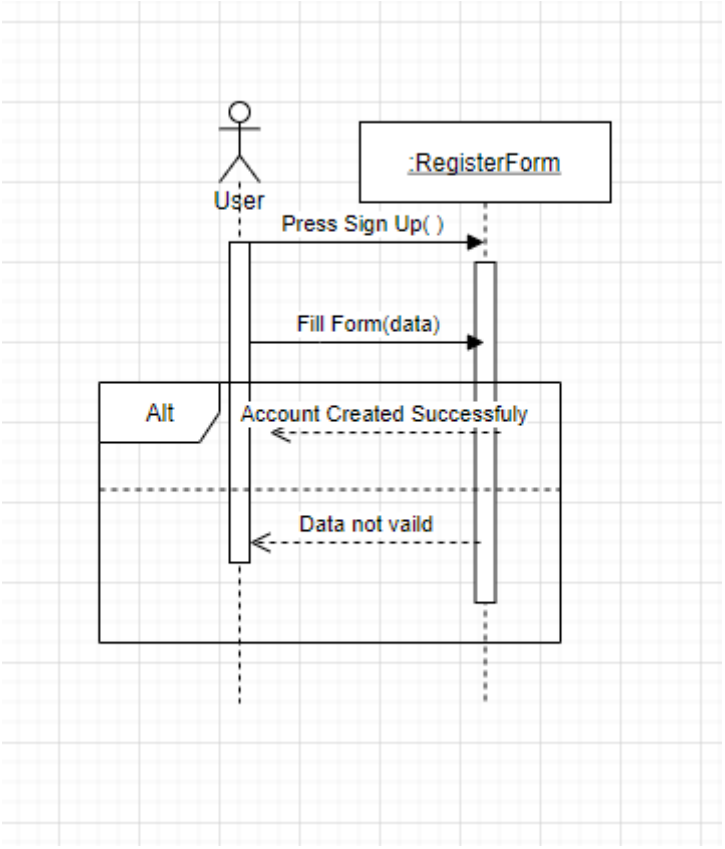


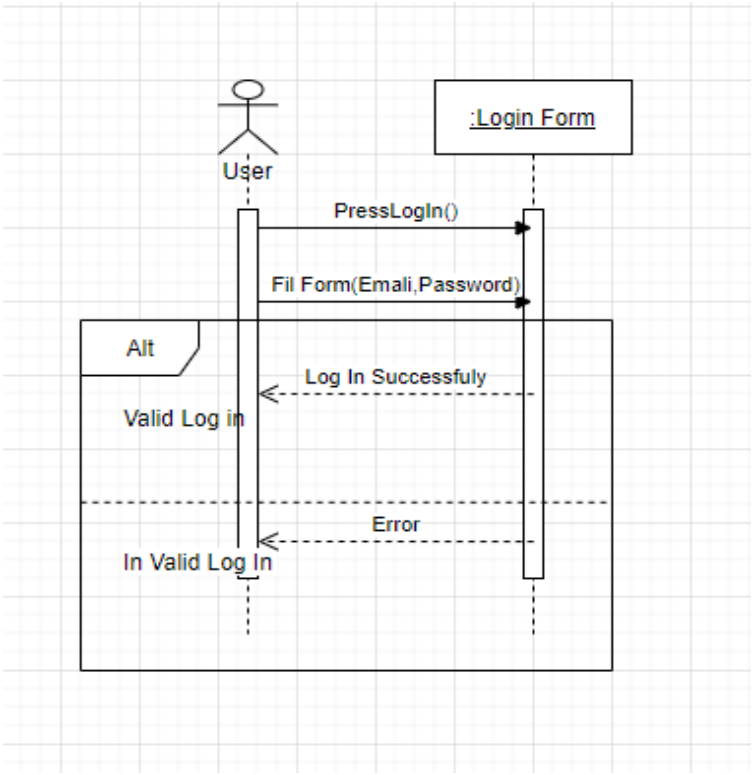
Figure 11 Log Out

Sequence Diagram:

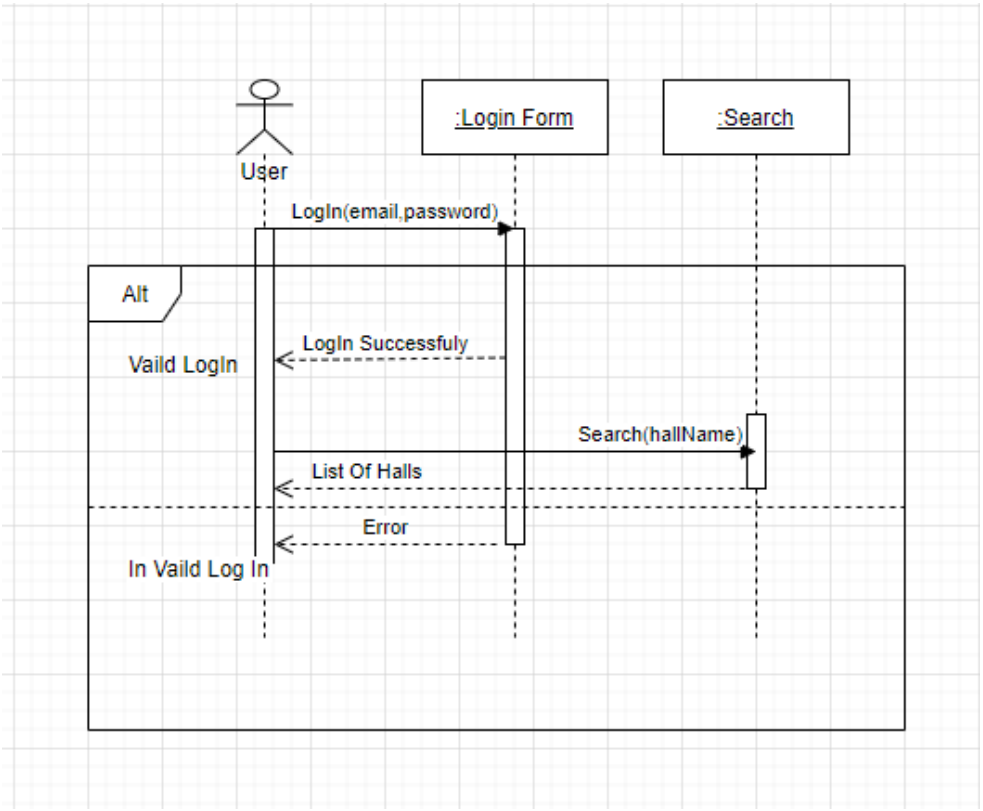
Use Case 1:



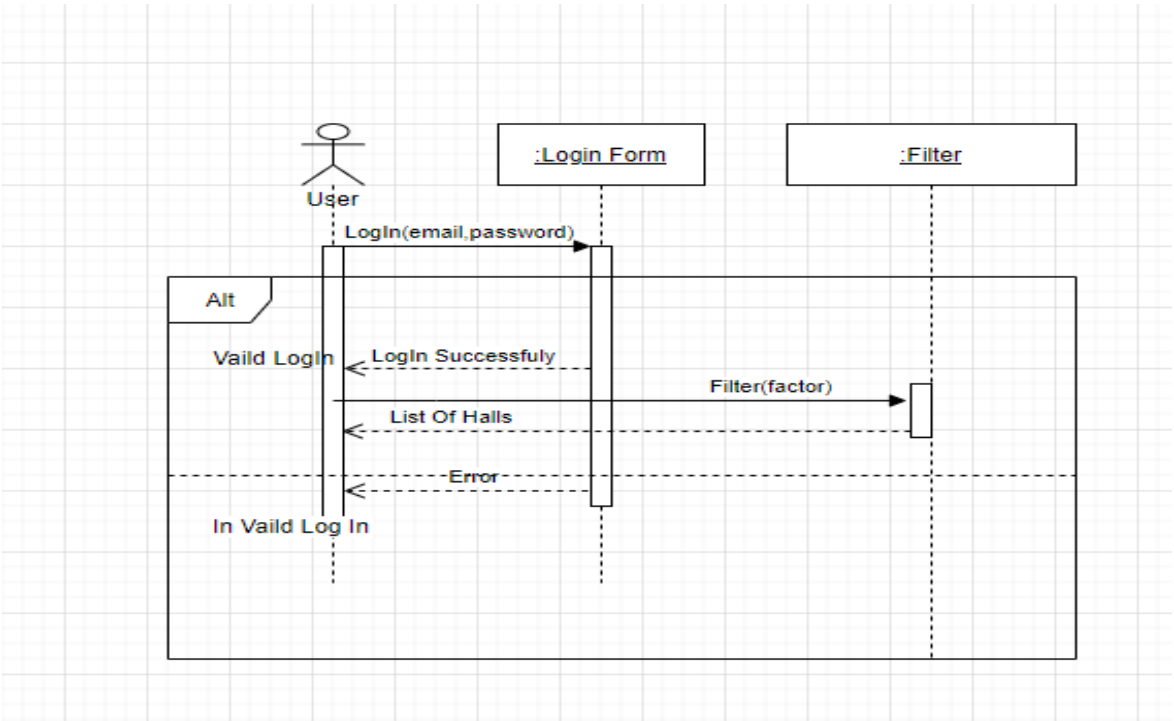
Use Case 2:



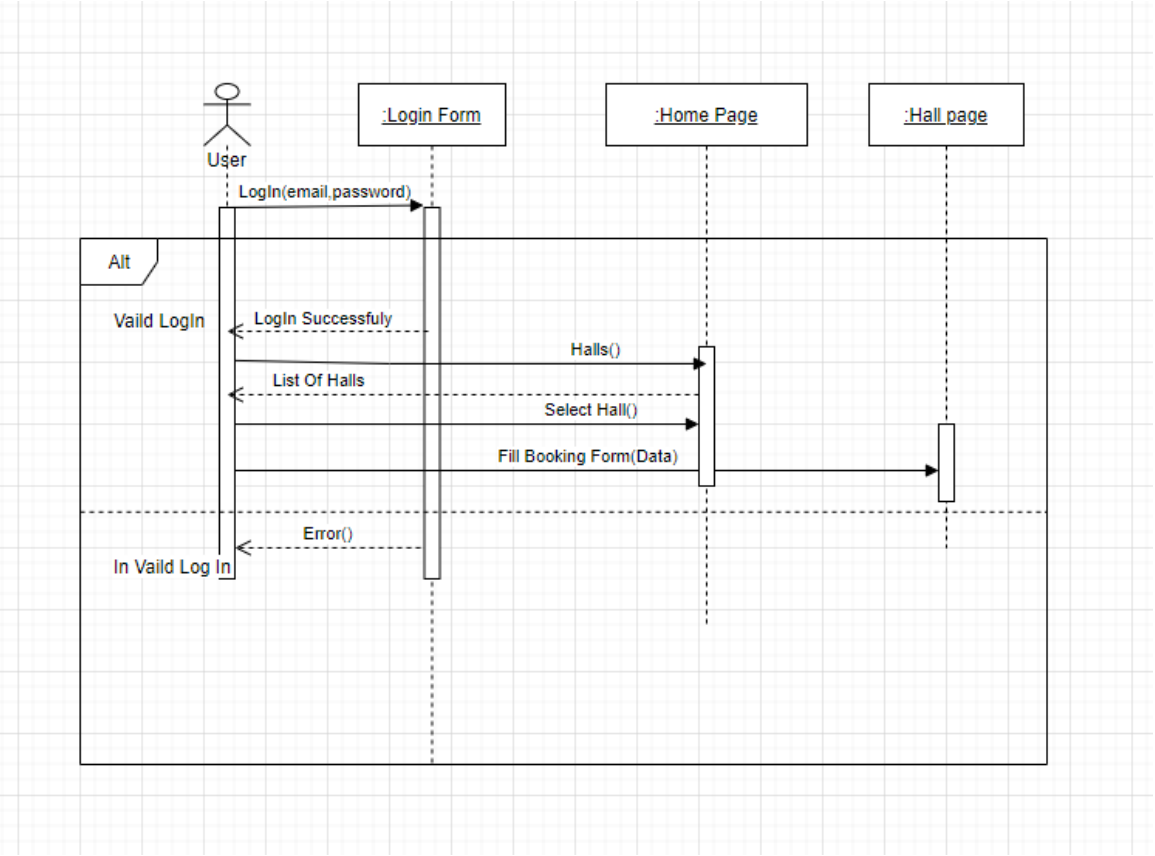
Use Case 3:



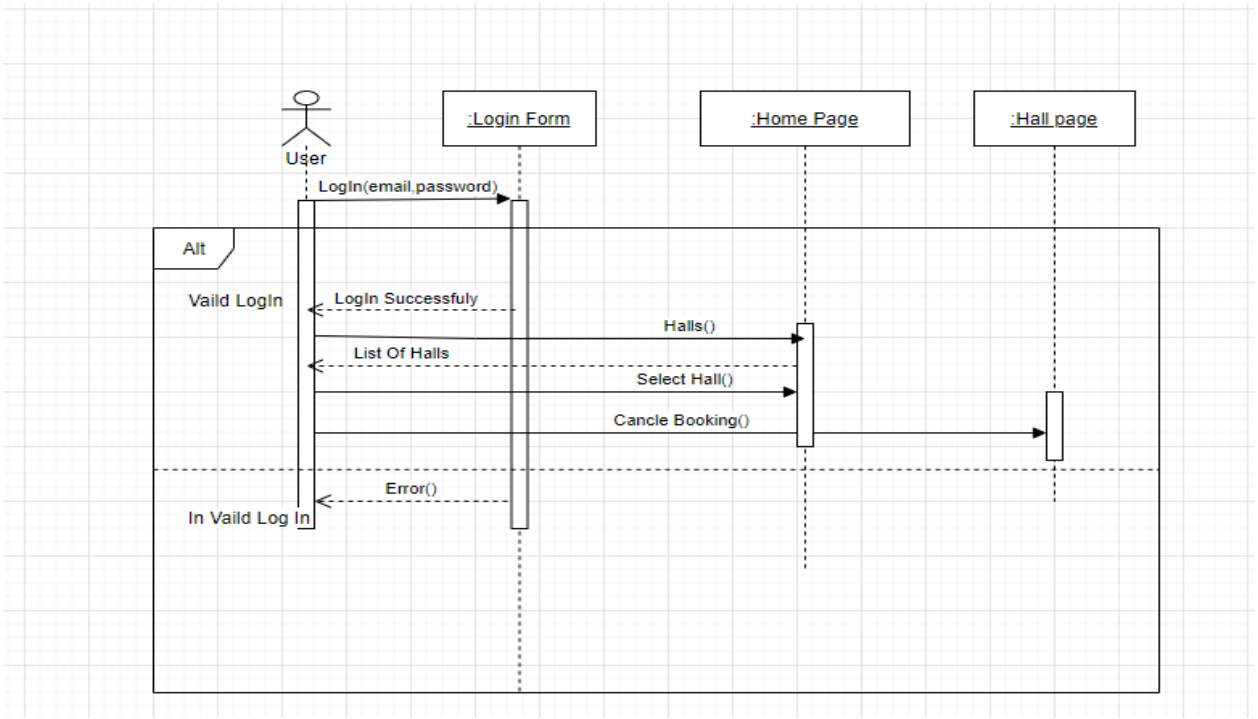
Use Case 4:



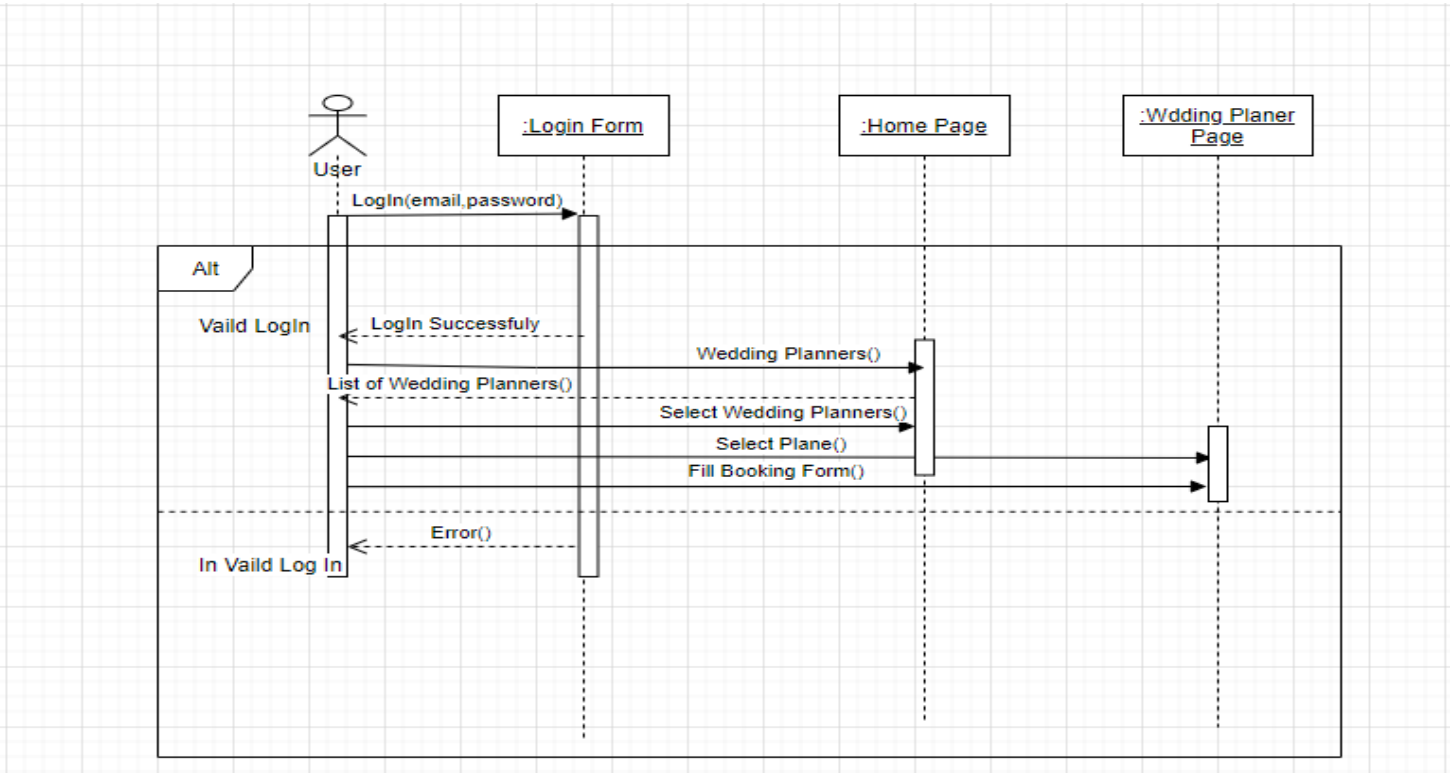
use Case5:


:Login Form
:Home Page
:Hall page
User
LogIn(email,password)
Alt
Vaild LogIn
LogIn Successfuly
Halls()
List Of Halls
Select Hall()
Fill Booking Form(Data)
Error()
In Vaild Log In

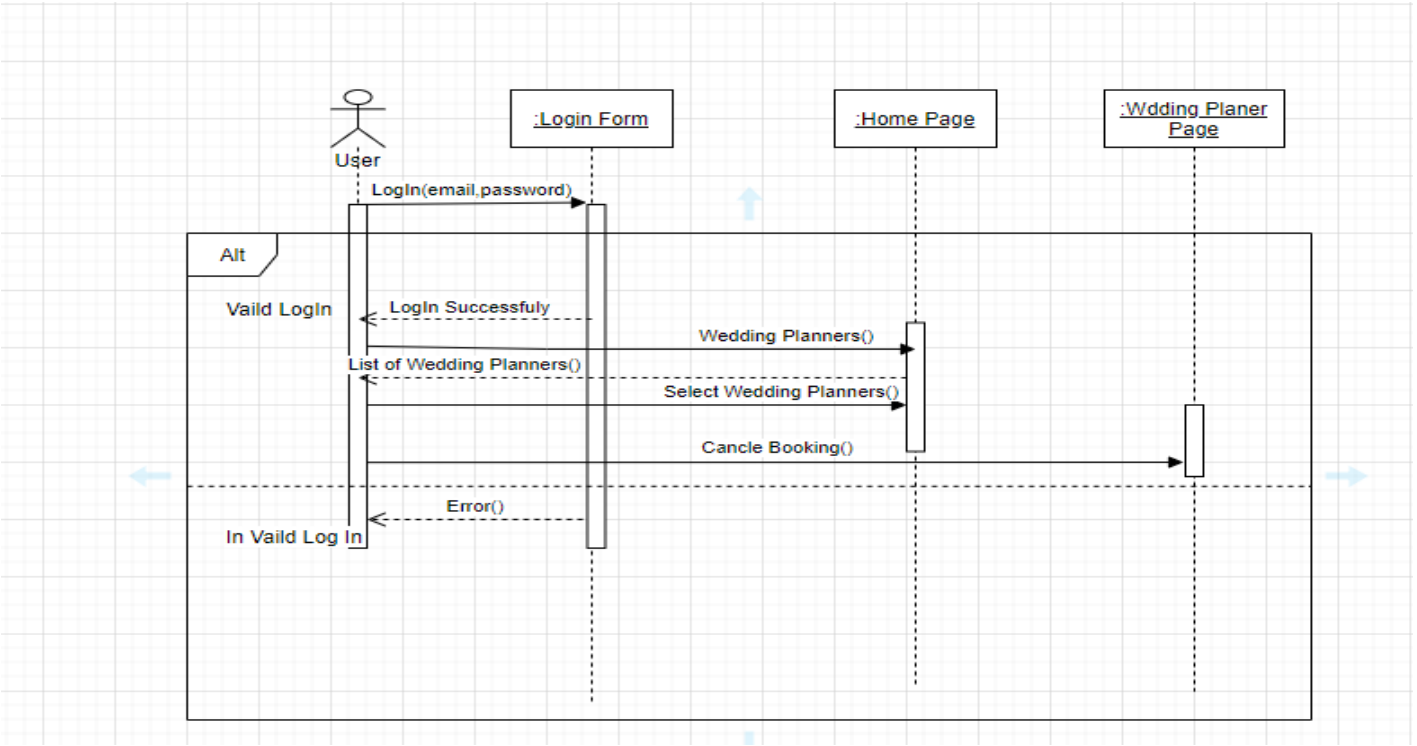
Use Case 6:


:Login Form
:Home Page
:Hall page
User
LogIn(email,password)
Alt
Vaild LogIn
LogIn Successfuly
Halls()
List Of Halls
Select Hall()
Cancle Booking()
Error()
In Vaild Log In

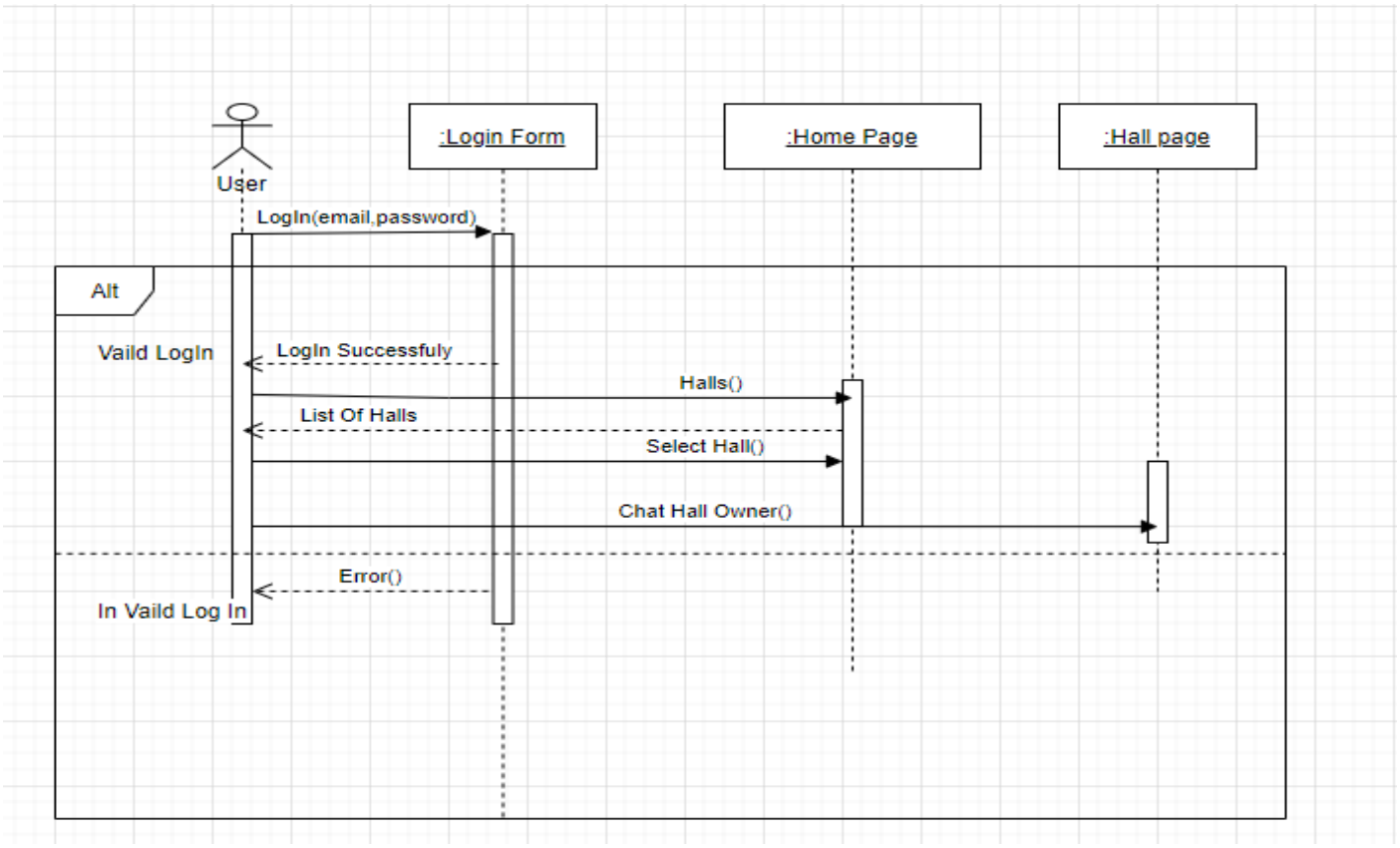
Use Case 7:



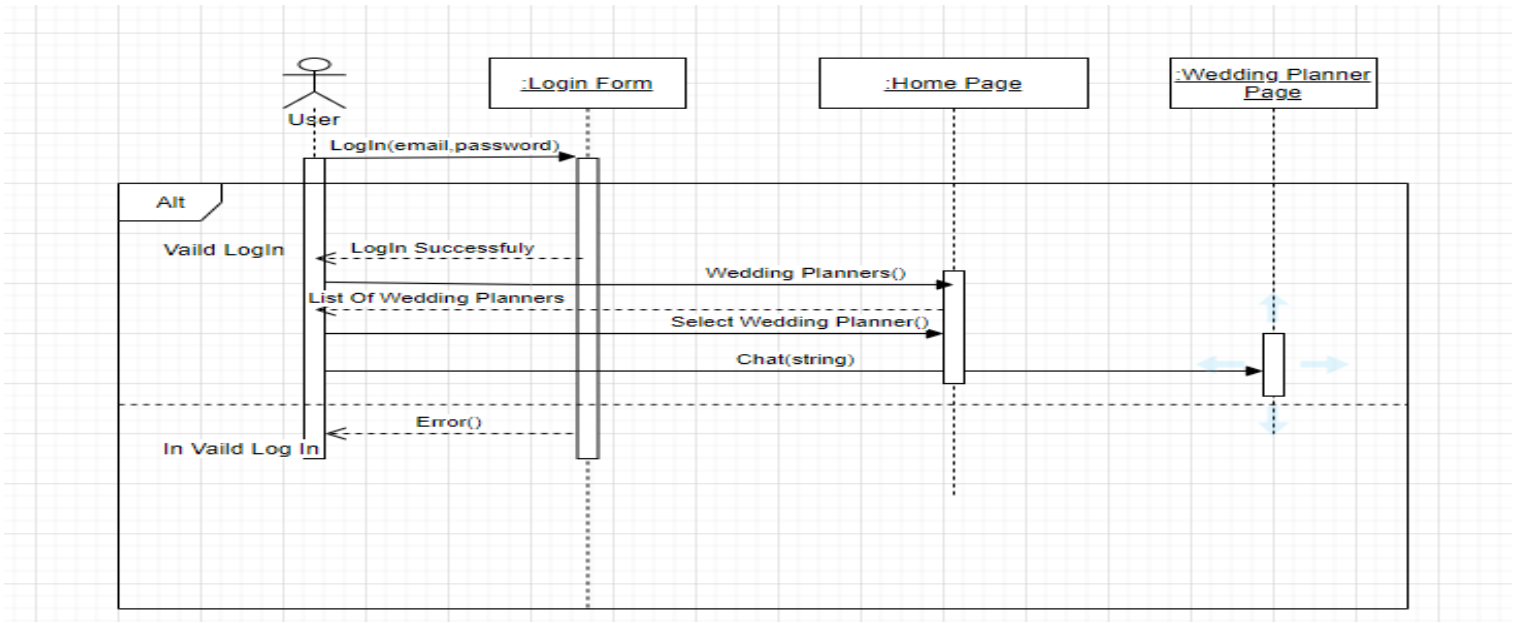
Use Case 8:



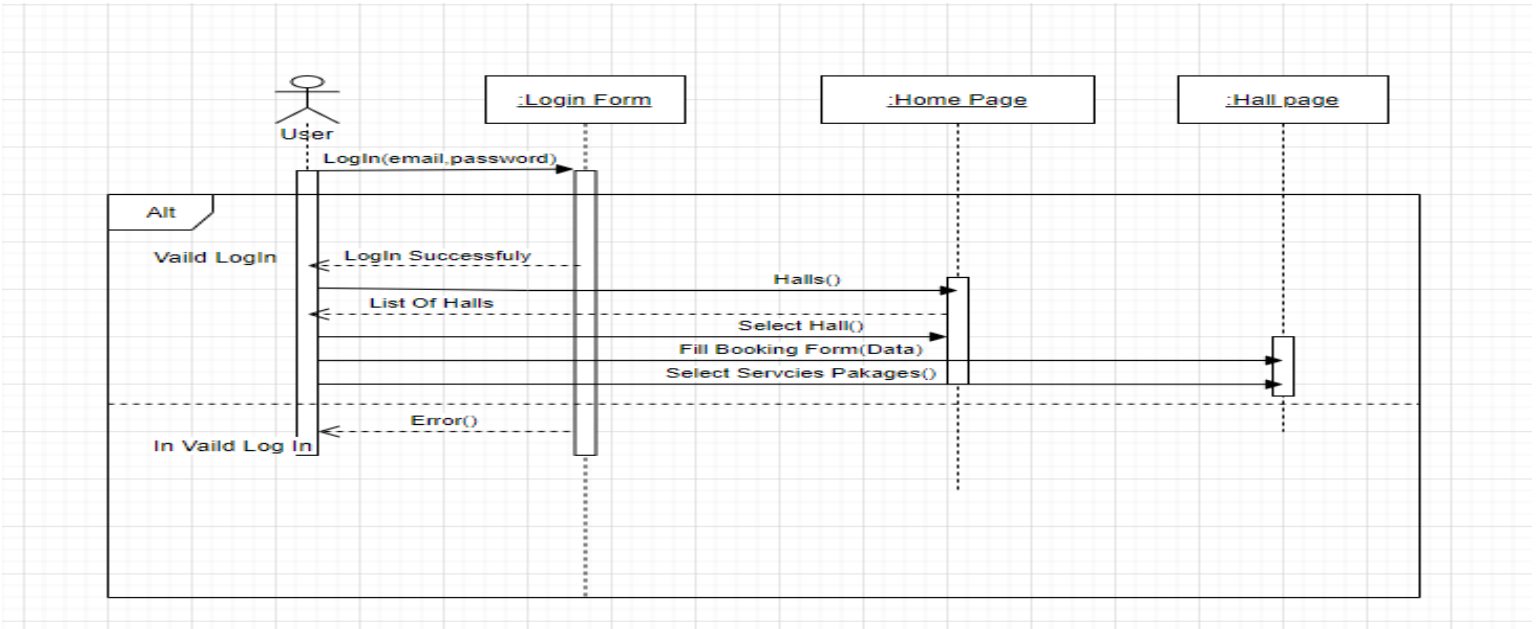
Use Case 9:



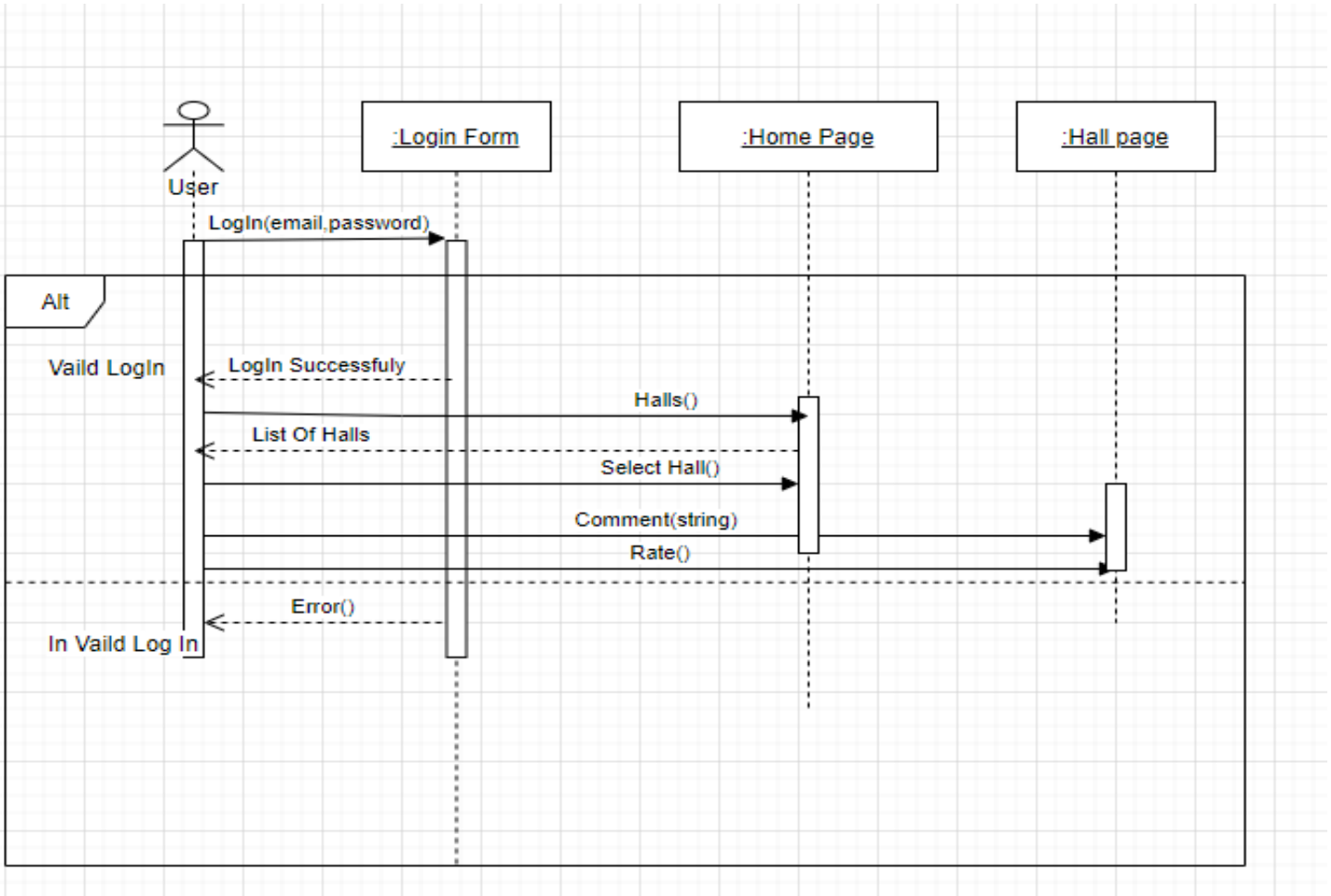
Use Case 10:



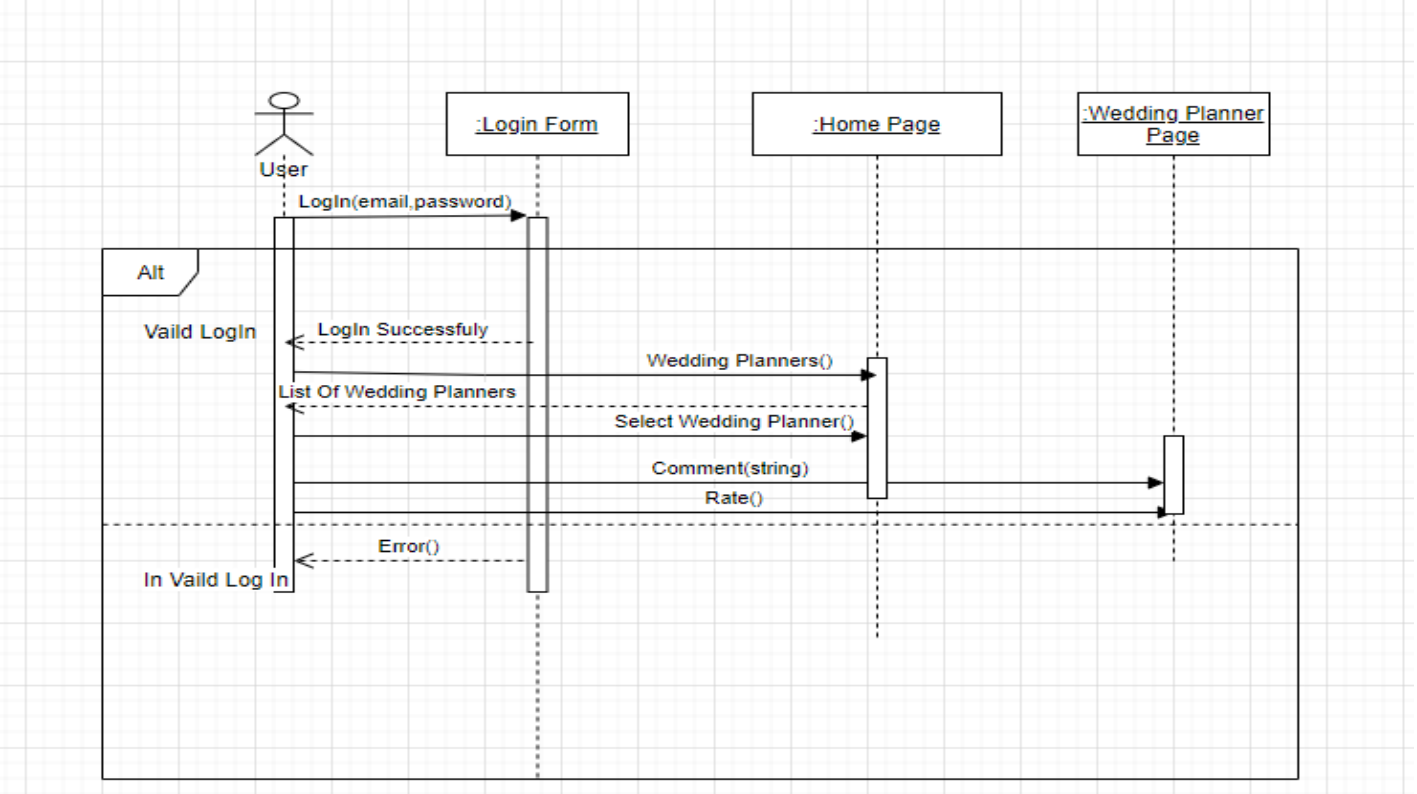
Use Case 11:



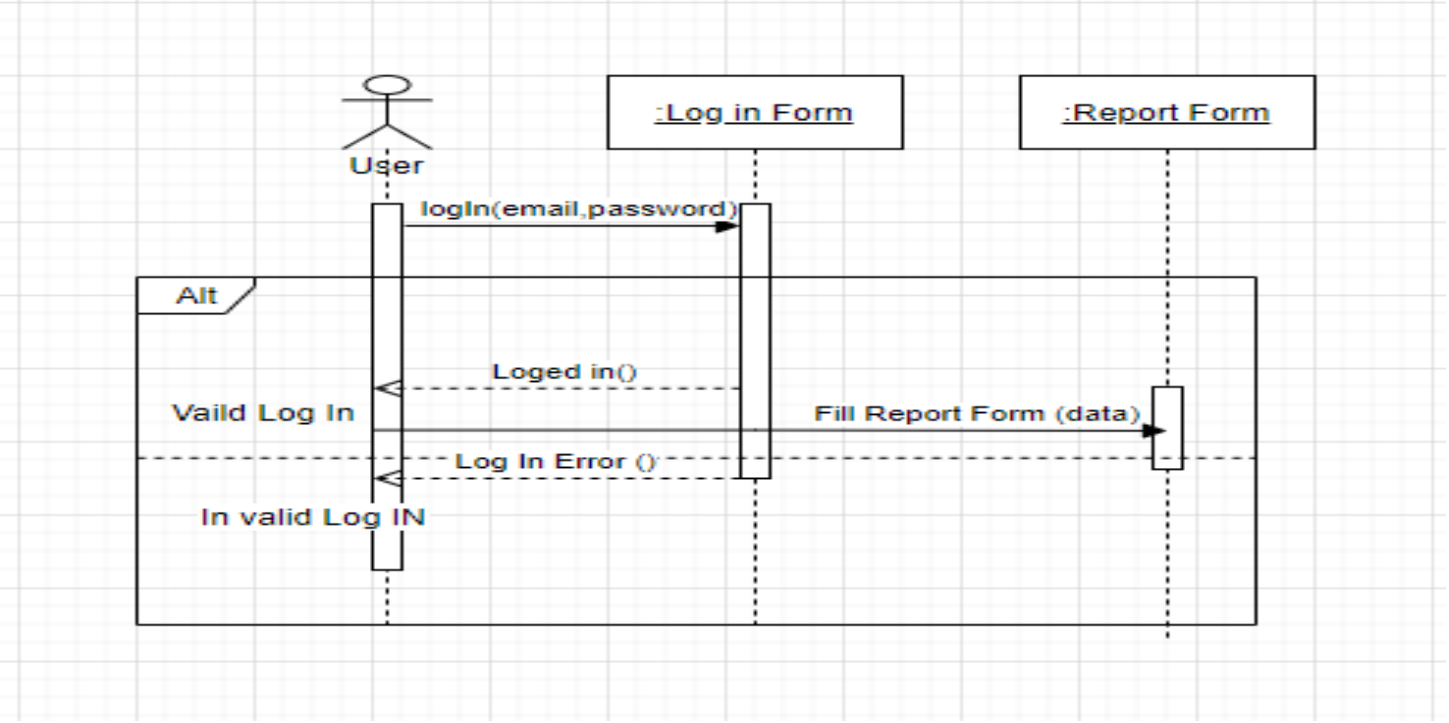
Use Case 12:



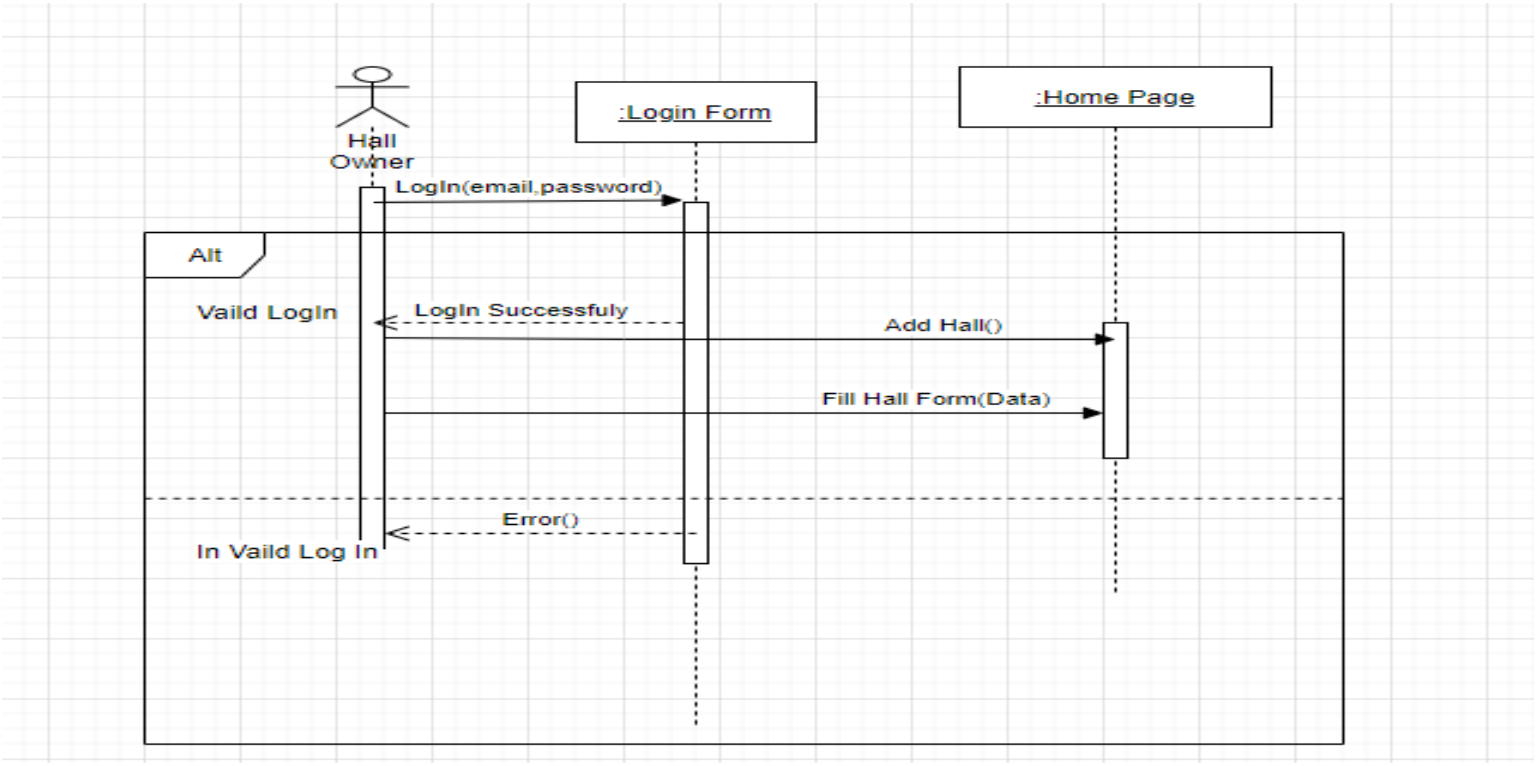
Use Case 13:



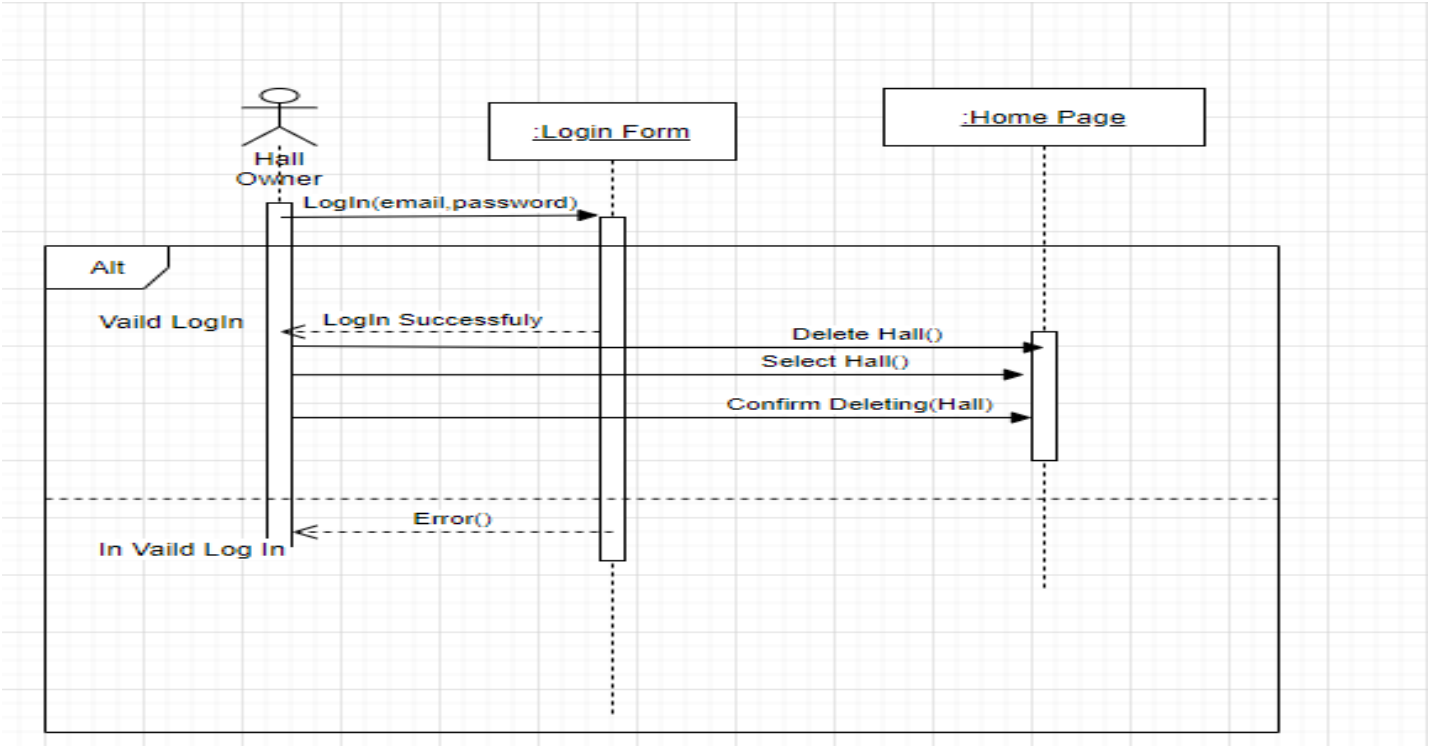
Use Case 14:



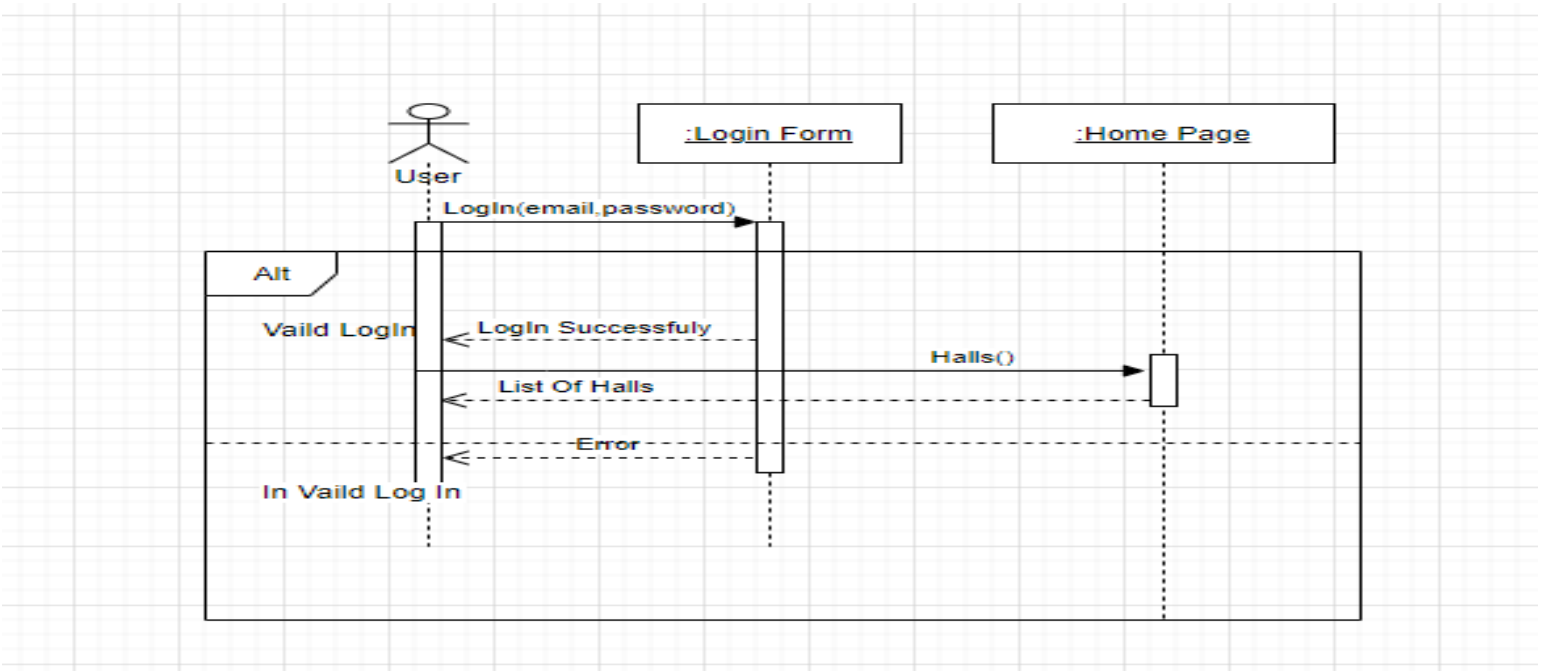
Use Case 15:



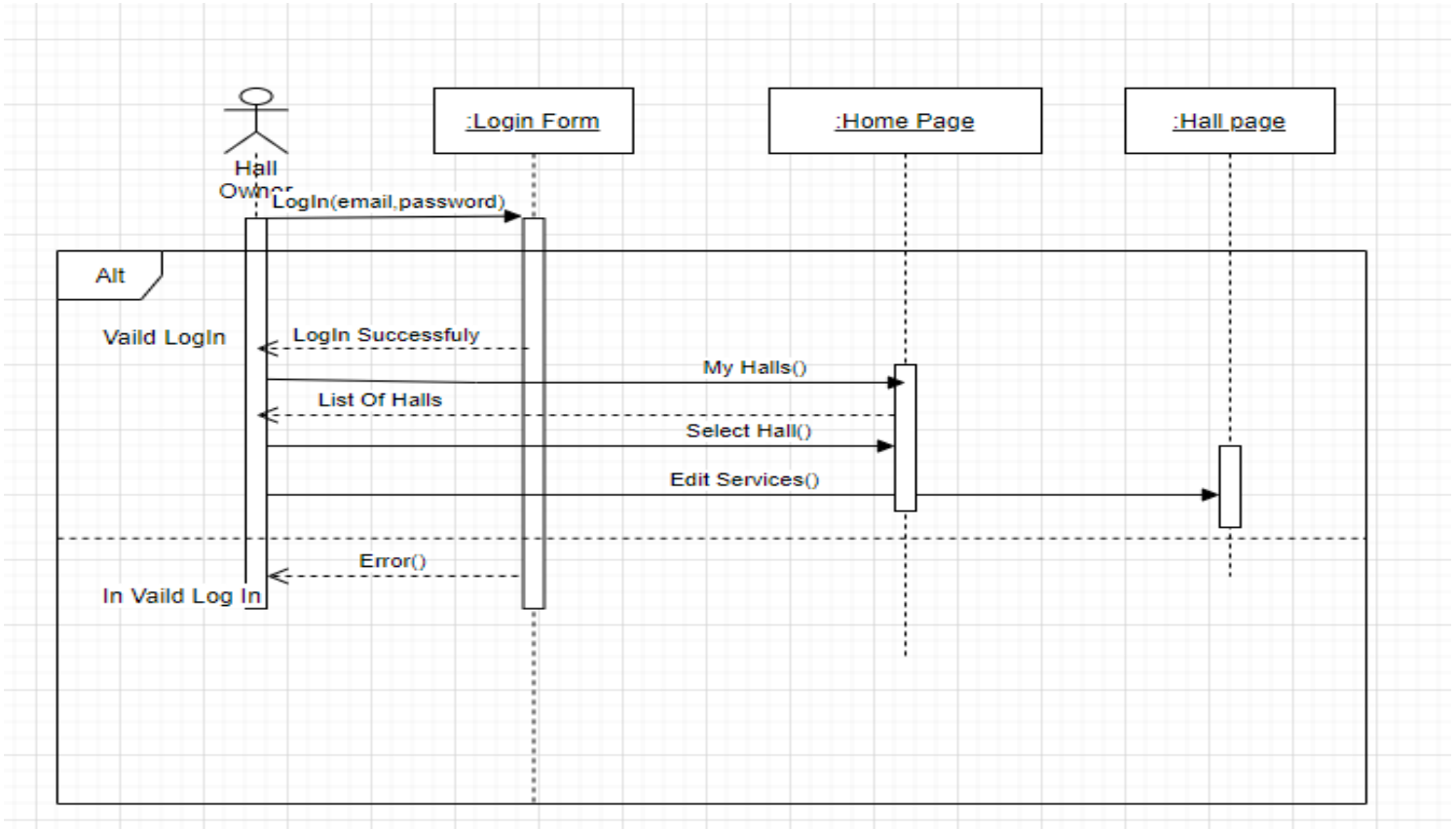
Use Case 16:



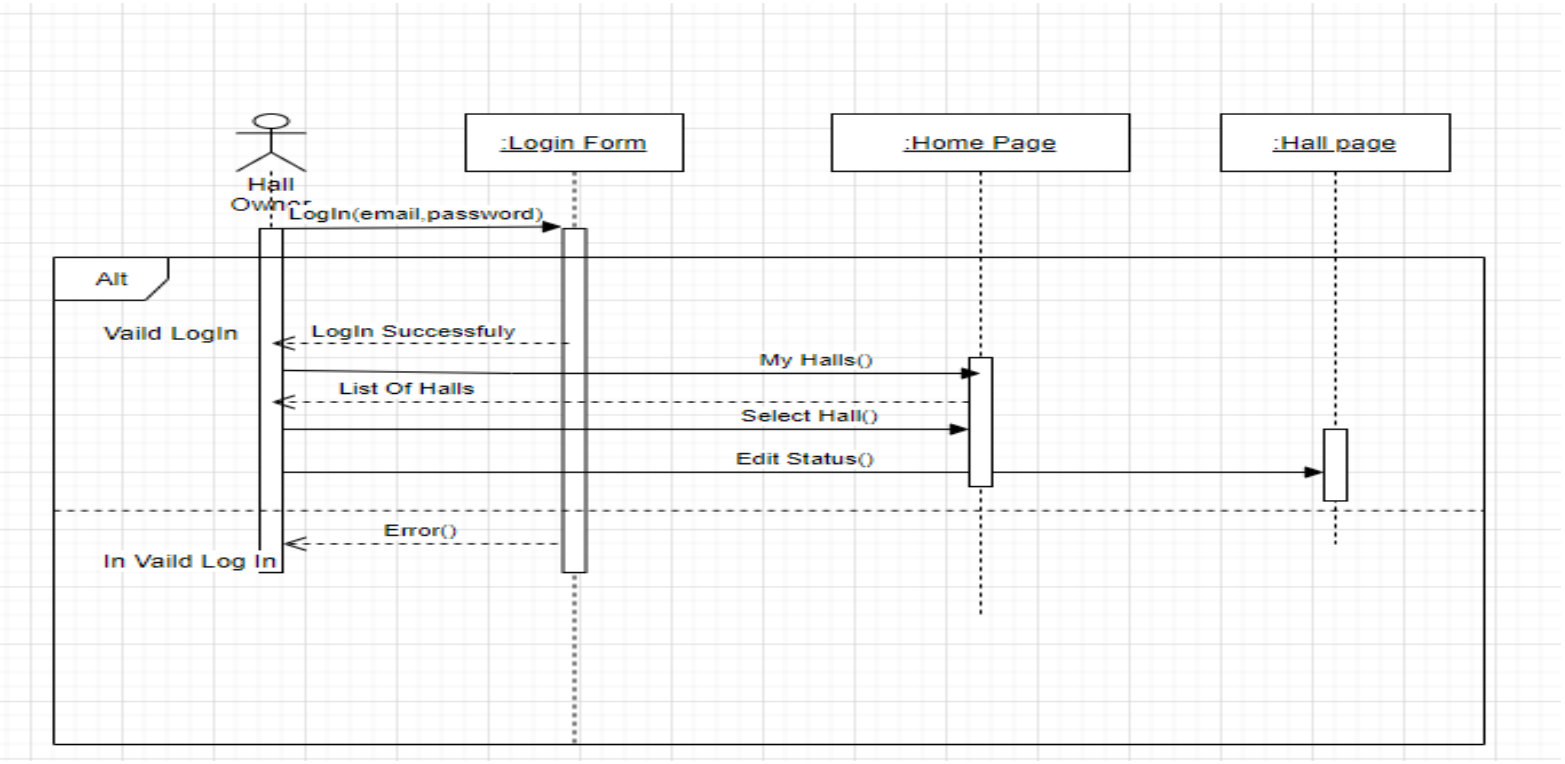
Use Case 17:



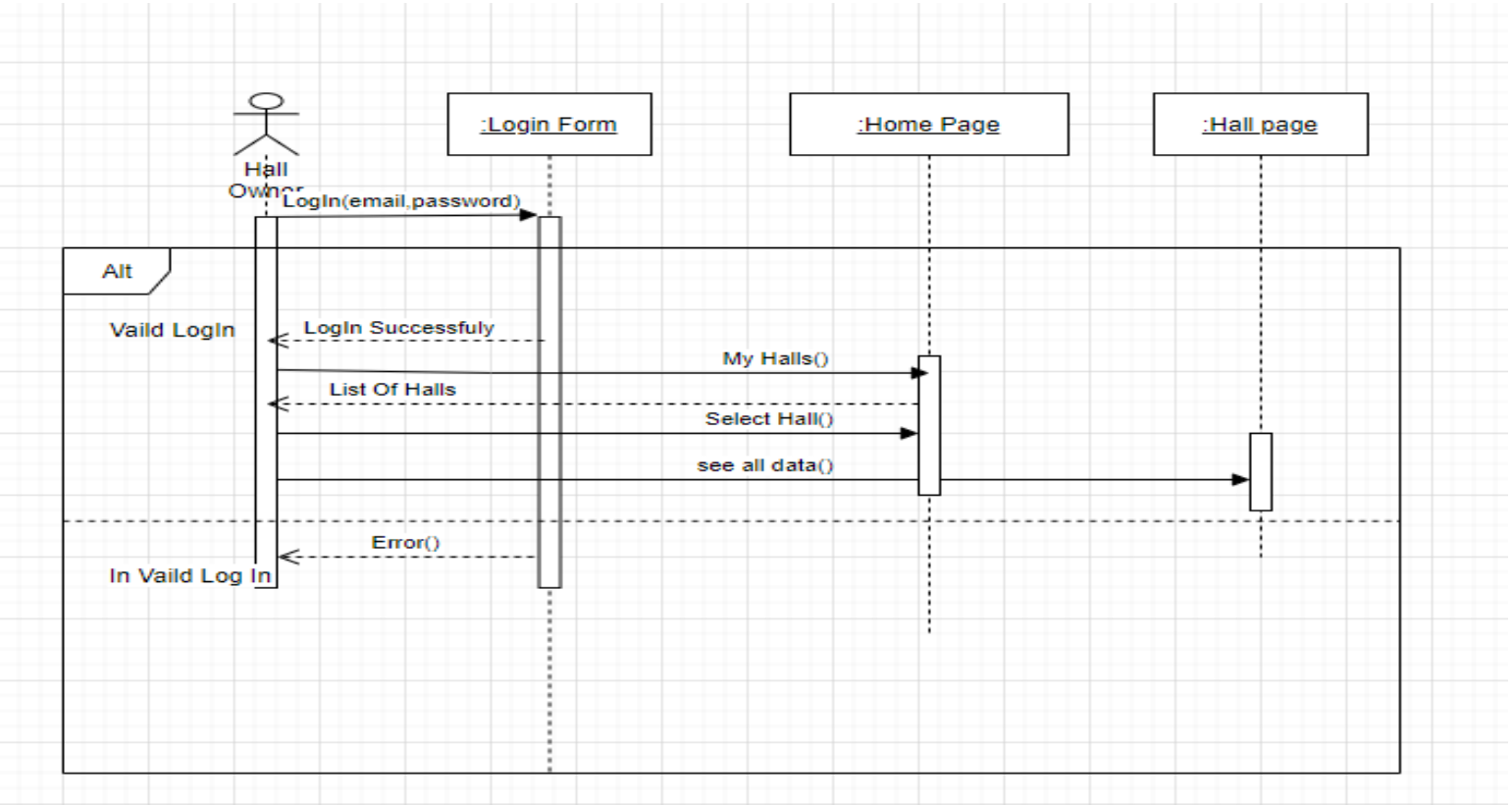
Use Case 18:



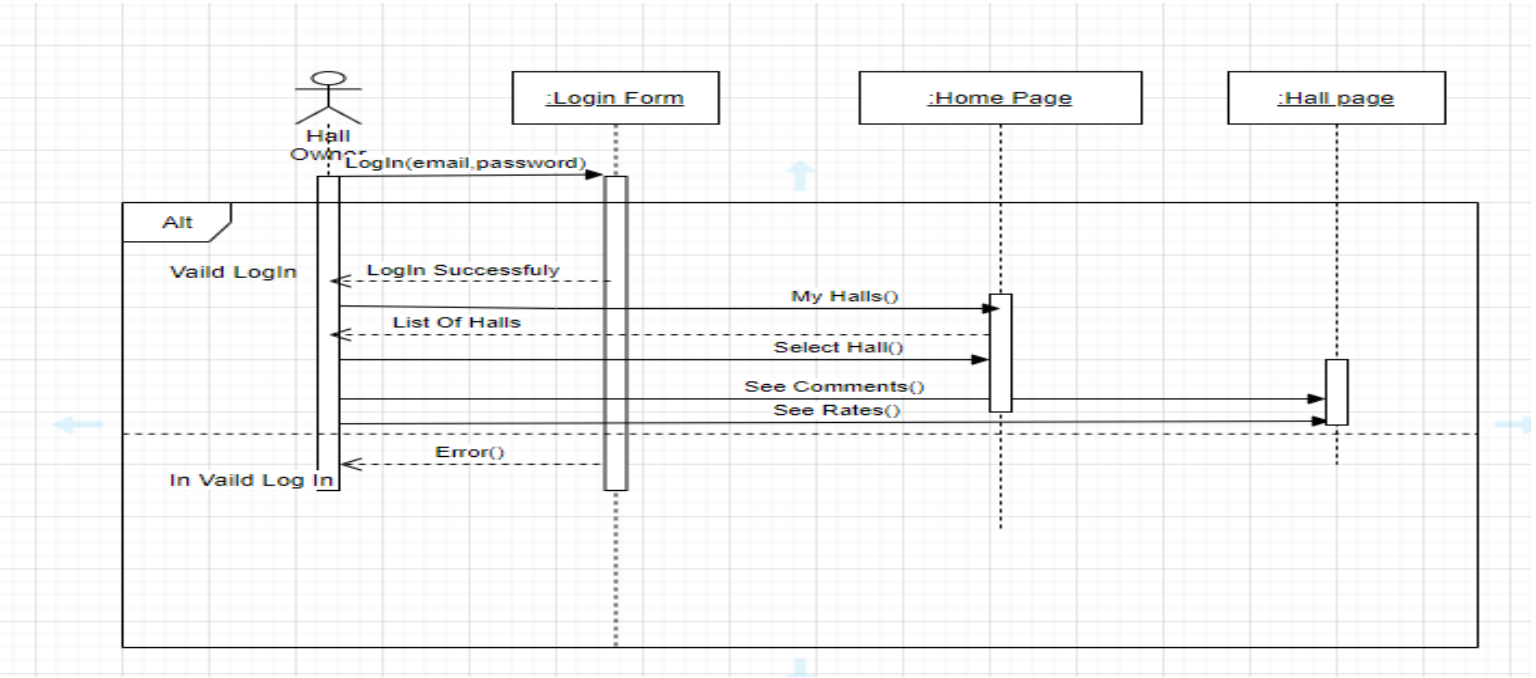
Use Case 19:



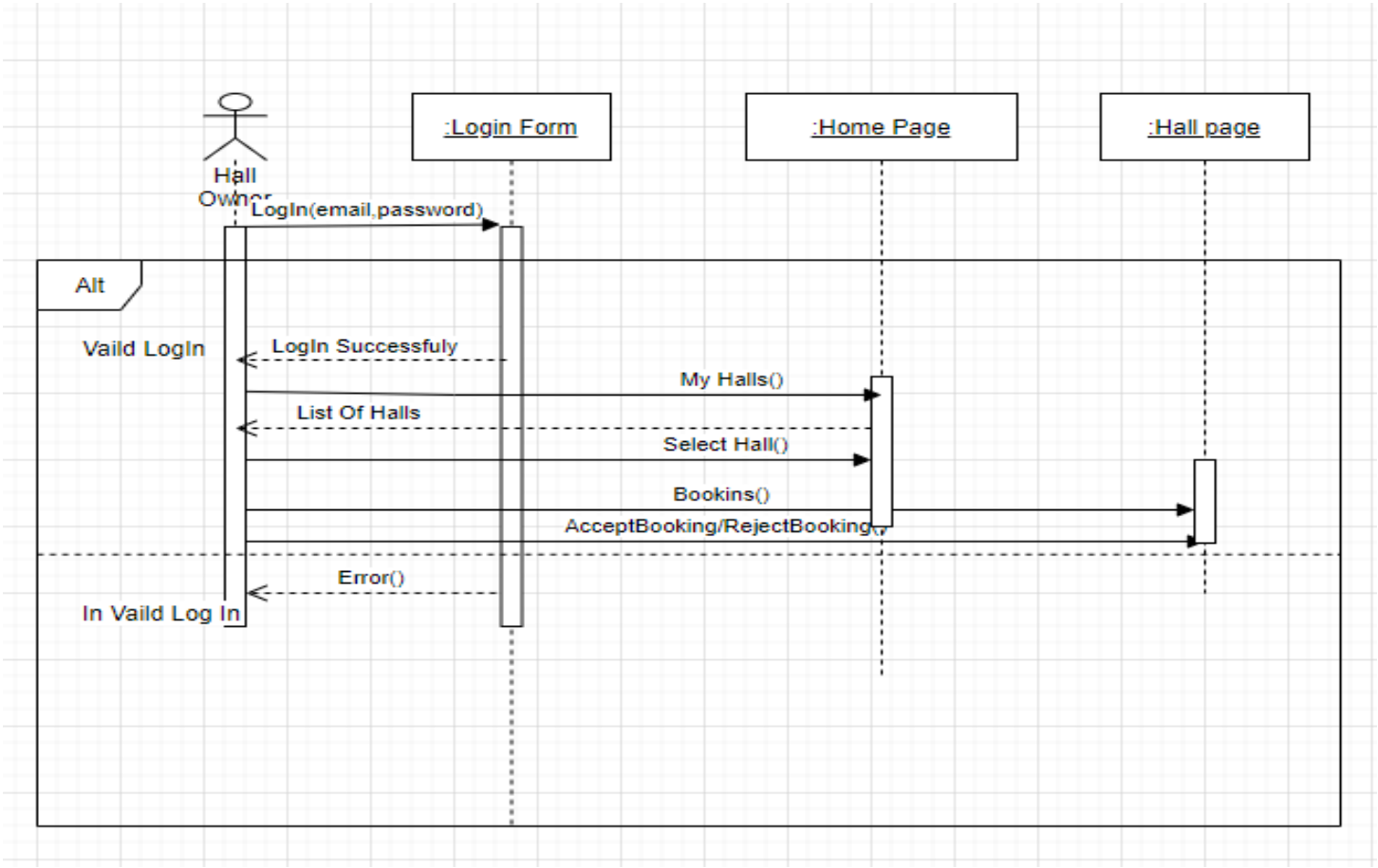
Use Case 20:



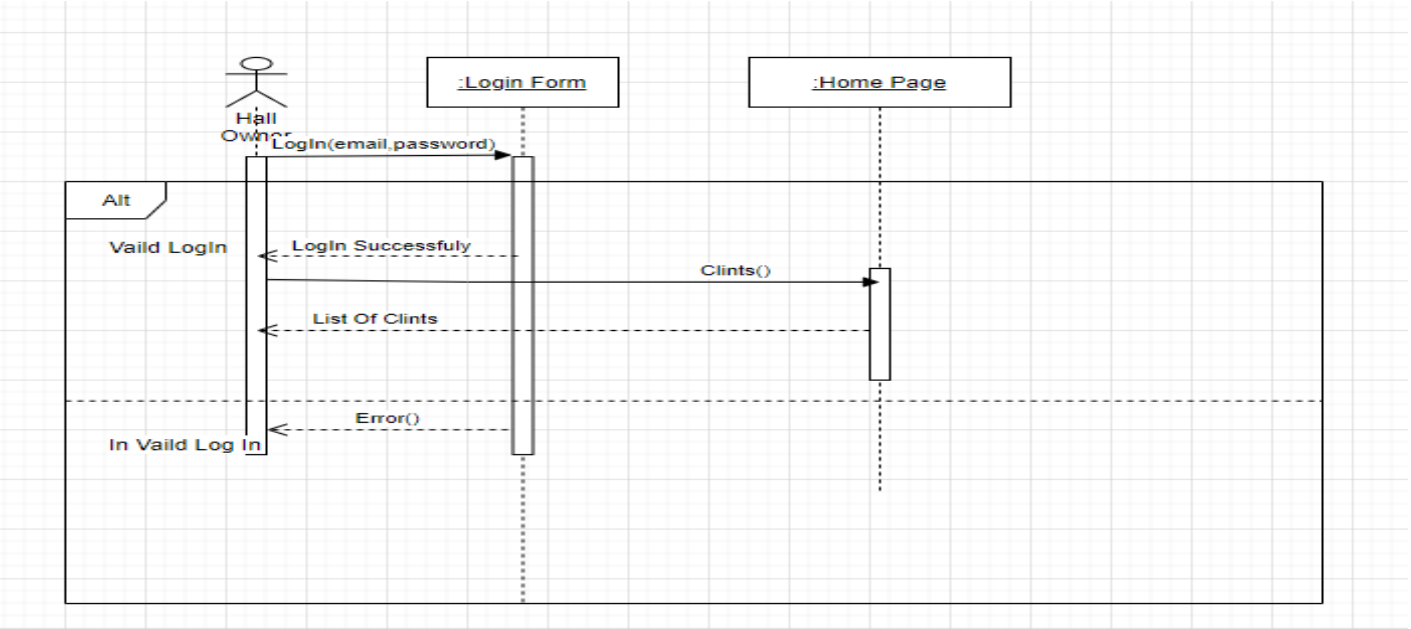
Use Case 21:



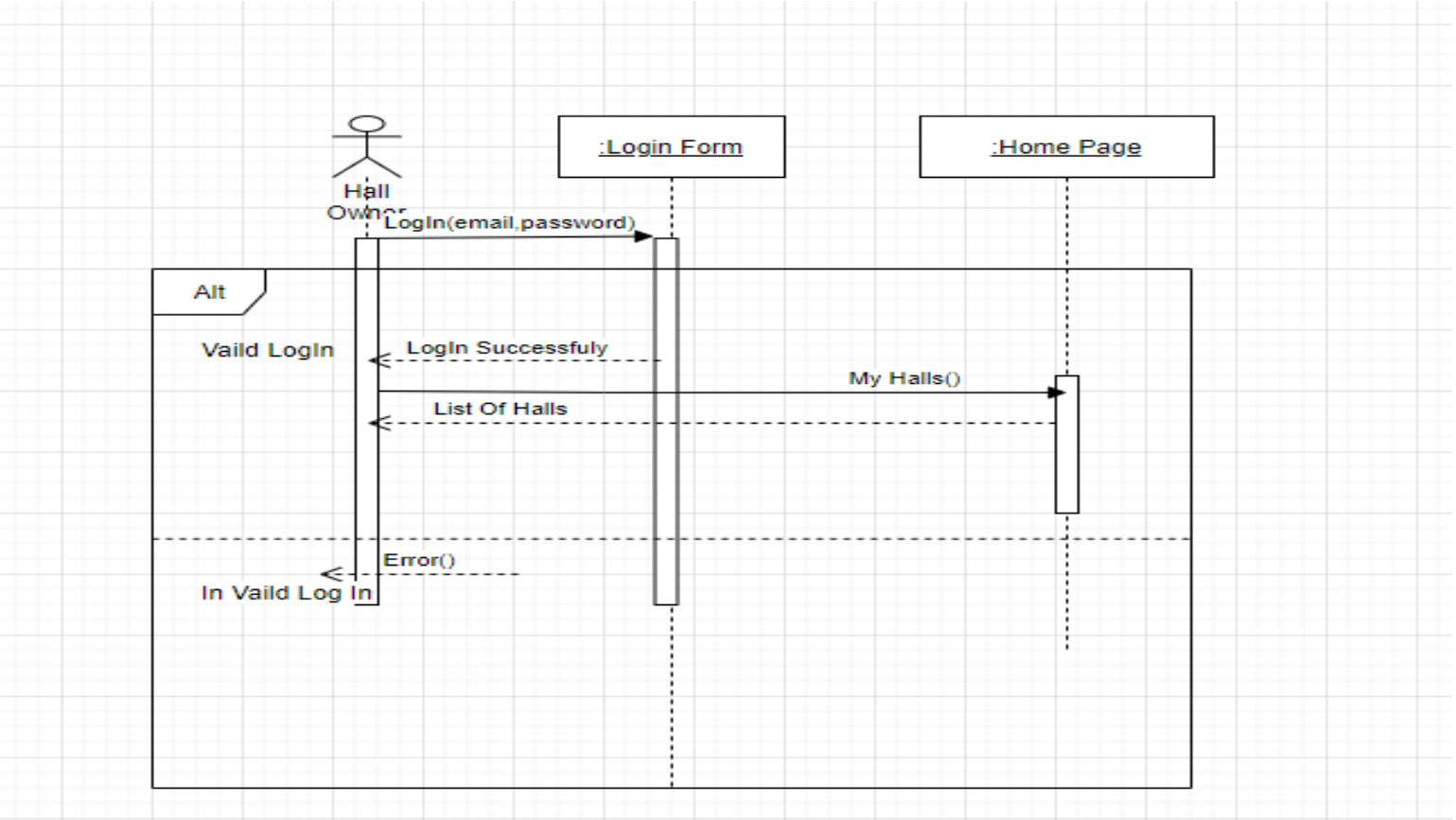
Use Case 22:



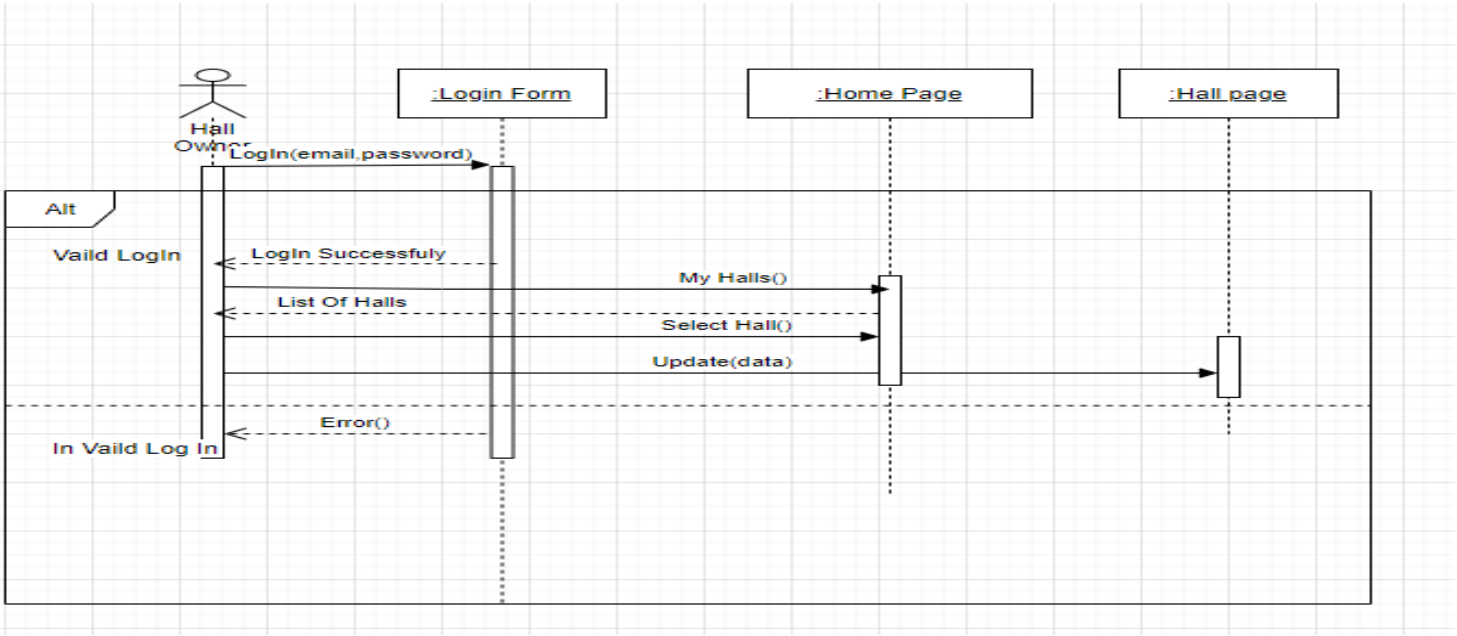
Use Case 23:



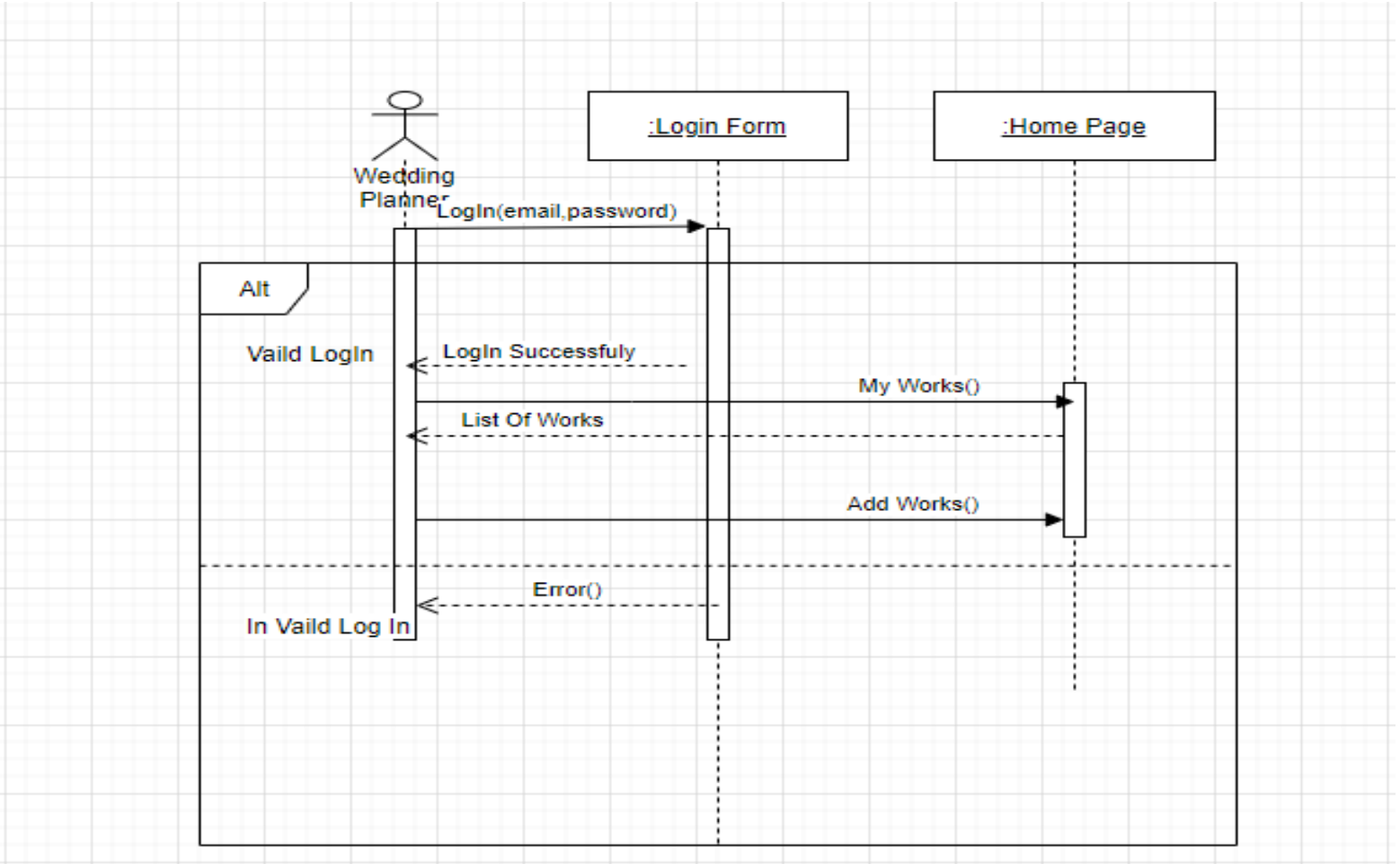
Use Case 24:



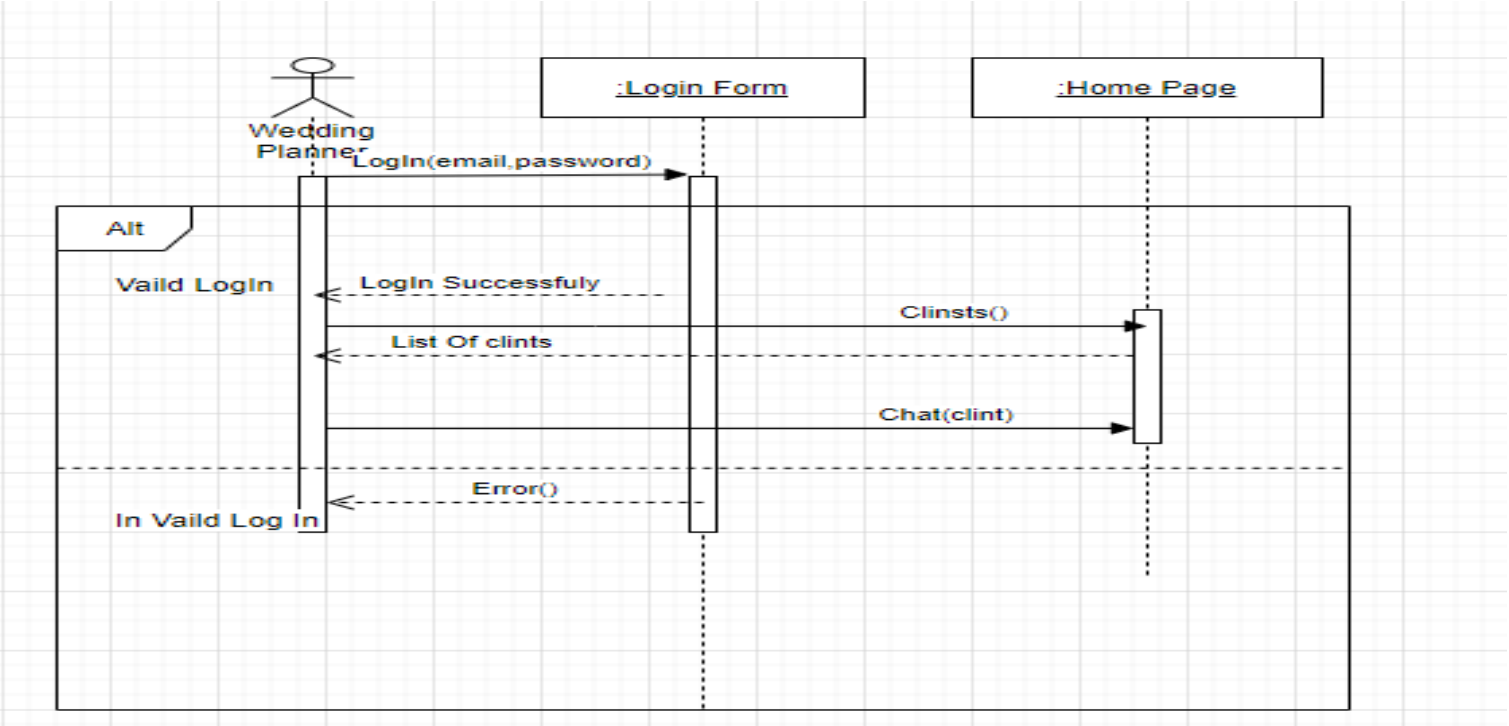
Use Case 25:



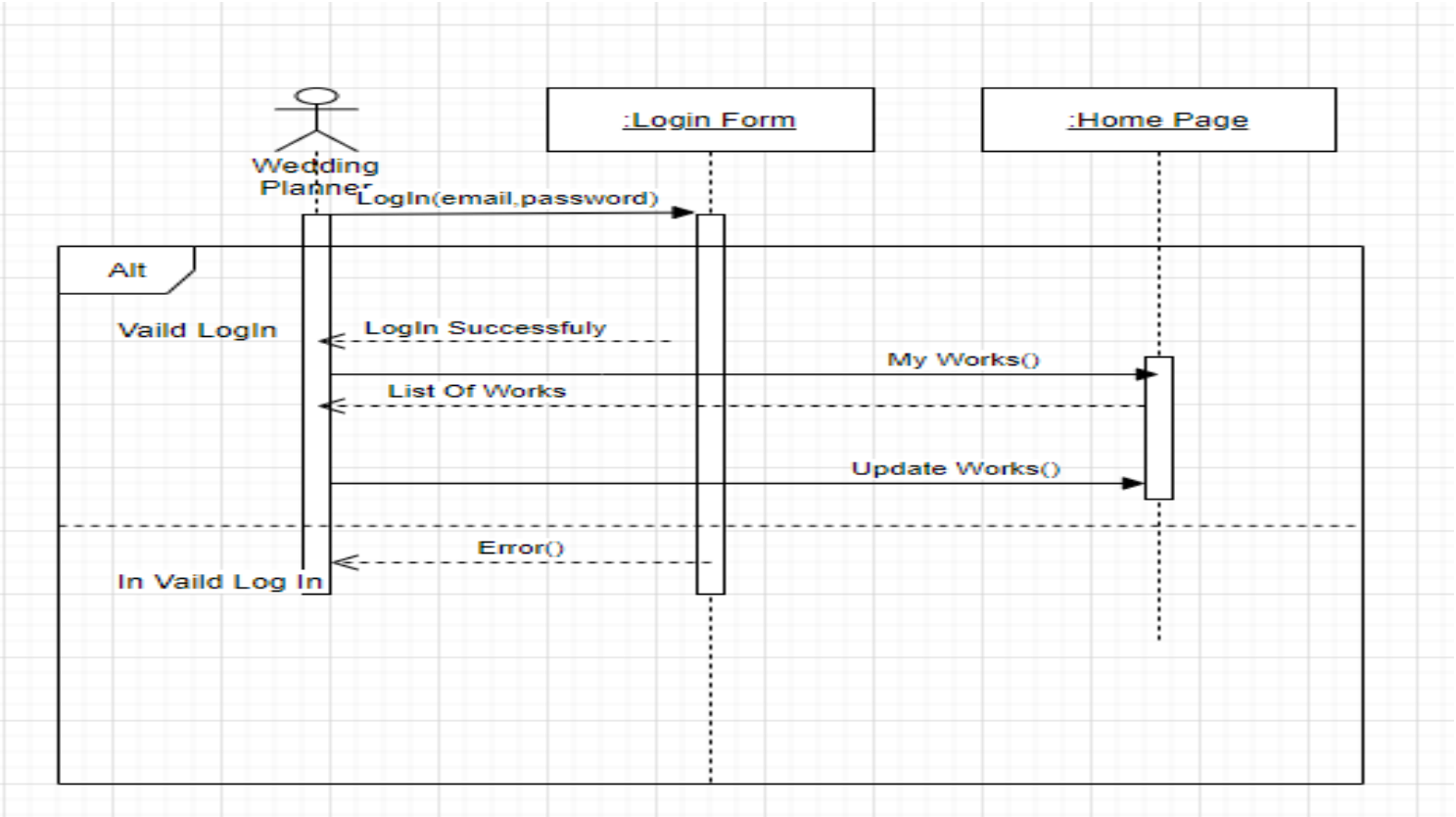
Use Case 26:



Use Case 27:



Use Case 28:



Chapter 4:
Implementation, Experimental Setup

4.1 Software Tools:

JavaScript is a popular high-level programming language that is primarily used for developing dynamic and interactive web applications. It is a versatile language that can be run on both the client-side (web browsers) and the server-side (using Node.js). JavaScript plays a crucial role in web development, allowing developers to enhance the functionality and interactivity of websites. JavaScript is continuously evolving, with regular updates and new features being added to the language. It has become a foundational language for web development, empowering developers to create dynamic, responsive, and interactive web applications.

ReactJS, often referred to as React, is an open-source JavaScript library for building user interfaces. It was developed by Facebook and is widely used for creating interactive and reusable UI components in web applications. React follows a component-based architecture, allowing developers to build complex UIs by breaking them down into smaller, reusable components. React has gained significant popularity due to its performance, modularity, and developer-friendly features. It is widely used in building single-page applications, progressive web apps, mobile apps (using React Native), and even desktop applications. With its focus on reusability and efficient UI rendering, react has revolutionized the way developers approach building user interfaces in JavaScript applications.

CSS (Cascading Style Sheets) is a styling language used to describe the visual appearance of web pages written in HTML or XML. It provides a set of rules and properties that control the layout, design, and presentation of elements on a web page. CSS allows you to define colours, fonts, sizes, margins, paddings, borders, backgrounds, and more. It also offers features for positioning elements, creating responsive designs, adding transitions and animations, and ensuring browser compatibility. With CSS, you can enhance the visual appeal and user experience of websites, making them more attractive and engaging.

HTML (Hypertext Markup Language) is the standard markup language used for creating web pages and applications. It provides the structure and content of a webpage, defining the elements and their arrangement on the page.

PHP PHP (Hypertext Pre-processor) is a popular server-side scripting language used for web development. It is primarily used for creating dynamic web pages and server-side applications. PHP code is embedded within HTML, allowing developers to mix PHP and HTML code seamlessly.

Laravel, Laravel, on the other hand, is a popular PHP framework known for its elegant syntax, expressive coding style, and robust features. Laravel follows the Model-View-Controller (MVC) architectural pattern, providing a structured and modular approach to building web applications.

Laravel's focus on developer productivity, elegant syntax, and rich feature set has made it a preferred choice for many PHP developers. It offers a well-rounded development experience and promotes best practices, allowing developers to build robust, scalable, and maintainable web applications efficiently.

XAMPP is a free and open-source cross-platform software package that provides a complete solution for setting up a local web server environment. The name "XAMPP" is an acronym for the components it includes Apache, MariaDB (formerly MySQL), PHP, and Perl.

XAMPP is a popular choice for developers who want a convenient and all-in-one solution for setting up a local web server environment. It provides the necessary components and tools to create, test, and deploy web applications locally, making it easier to develop and troubleshoot websites and web applications.

Visual Studio Code is a popular source-code editor developed by Microsoft. It is a lightweight and highly customizable tool that supports a wide range of programming languages and offers a variety of features to enhance developer productivity.

VS Code has gained popularity among developers due to its versatility, performance, and extensive feature set. Its combination of a lightweight yet powerful editor, customizable interface, and robust extension ecosystem makes it a favoured choice for coding in various programming languages and frameworks.

Postman is a popular API development and testing tool used by developers to simplify the process of building, testing, and documenting APIs (Application Programming Interfaces). It provides a user-friendly interface that allows developers to send requests to APIs, receive responses, and analyse the data exchanged between the client and server.

Postman simplifies the process of API development, testing, and collaboration, providing developers with a comprehensive set of tools to interact with APIs. Its

user-friendly interface, extensive features, and integrations make it a valuable tool for API-centric development workflows.

MySQL is an open-source relational database management system (RDBMS) that is widely used for managing and storing structured data. It is one of the most popular databases in the world and is known for its performance, reliability, and ease of use.

MySQL is widely used in web applications, content management systems, e-commerce platforms, data-driven applications, and various other scenarios that require efficient and reliable data storage and management. Its popularity, extensive community support, and rich feature set make it a preferred choice for many developers and organizations.

Figma is a cloud-based design and prototyping tool used for creating user interfaces, interactive prototypes, and collaborative design workflows. It is popular among designers and design teams for its powerful features and real-time collaboration capabilities. Figma can be accessed through a web browser or a desktop application and is available on both macOS and Windows platforms.

Figma's collaborative nature, powerful design features, and cross-platform accessibility have made it a popular choice for designers and design teams. Its cloud-based approach eliminates the need for manual file syncing and provides a seamless experience for collaboration, design iteration, and design handoff.

Git is a distributed version control system that allows developers to track changes to their codebase,

collaborate with others and manage different versions of their projects. It was created by Linus Torvalds,

the same person who created the Linux operating system.

GitHub, on the other hand, is a web-based platform built on top of Git.

It provides hosting services for Git repositories, along with additional features and functionalities to enhance collaboration and project management.

While Git is the underlying version control system, GitHub provides a platform for hosting Git repositories, collaboration, and project management. It has become one of the most popular platforms for developers to share and collaborate on code, particularly in the open-source community.

Python is a versatile high-level programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability and a clean syntax, making it an excellent choice for beginners and experienced developers alike. Python's simplicity, versatility, and large ecosystem have contributed to its widespread adoption and popularity. Its ease of use and readability make it an excellent choice for beginners, while its extensive capabilities make it a powerful tool for experienced developers tackling complex projects.

4.2 Software architecture

-MVC

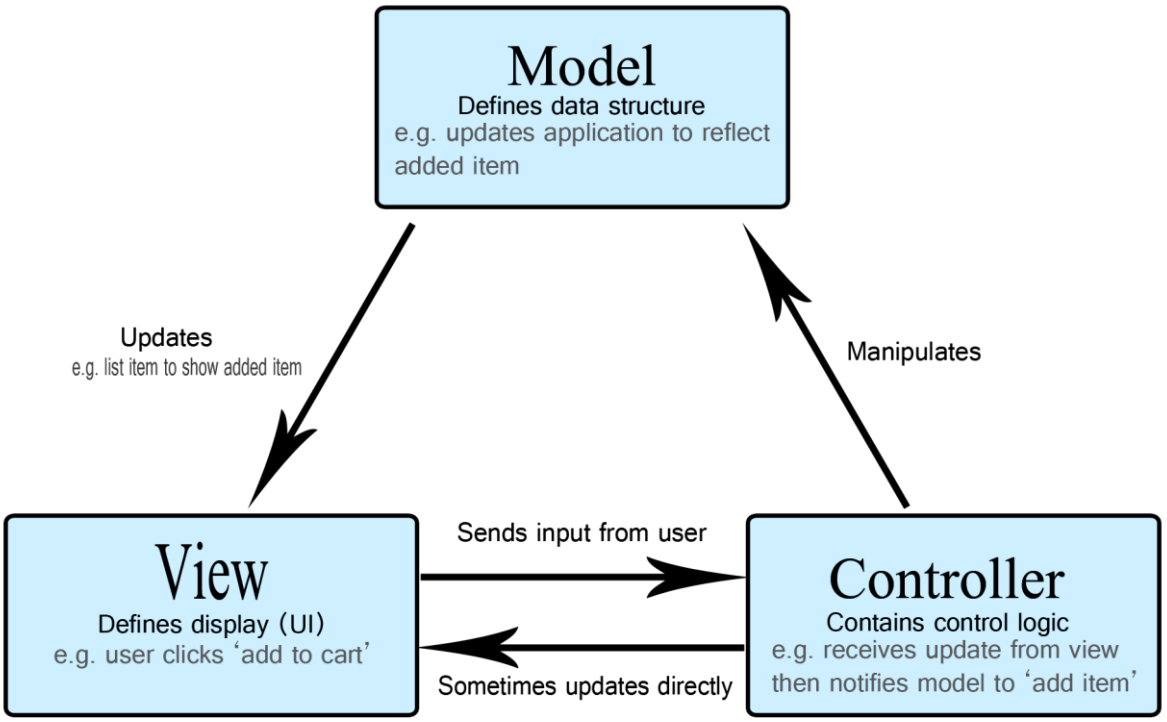


Figure 12 MVC

-MVC stands for Model-View-Controller, which is a software architectural pattern used in the development of web applications. The MVC pattern separates the application into three interconnected components, namely:

1. Model: This component represents the data and business logic of the application. It is responsible for managing data storage, retrieval, and manipulation.

2. View: This component represents the user interface of the application. It is responsible for displaying the data to the user and receiving user inputs.

3. Controller: This component acts as an intermediary between the Model and View components. It receives user inputs from the View

component, manipulates the data in the Model component as required, and updates the View component to reflect the changes.

-1-login

```php
public function switchLogin(Request $request)
{
    try {
        $rules = [
            'email' => 'required|email',
            'password' => 'required|string',
        ];
        $validator = Validator::make($request->all(), $rules);
        if ($validator->fails()) {
            $code = $this->returnCodeAccordingToInput($validator);
            return $this->returnValidationError($code, $validator);
        }

        $credentials = $request->only(['email', 'password']);

        if (Auth::guard('admin-api')->attempt($credentials)) {
            // User authenticated with admin guard
            return $this->loginAdmin($credentials);
        } elseif (Auth::guard('user-api')->attempt($credentials)) {
            // User authenticated with user guard
            return $this->loginUser($credentials);
        } elseif (Auth::guard('planner-api')->attempt($credentials)) {
            // User authenticated with planner guard
            return $this->loginPlanner($credentials);
        } elseif (Auth::guard('owner-api')->attempt($credentials)) {
            // User authenticated with owner guard
            return $this->loginOwner($credentials);
        }elseif (Auth::guard('supplier-api')->attempt($credentials)) {
            // User authenticated with owner guard
            return $this->loginSupplier($credentials);
        } else {
            // User dose not exit
            return response()->json([
                'message' => 'Email or Password Doesn`t Exist',

            ], 201);
        }
    } catch (\Exception $ex) {
        return $this->returnError($ex->getCode(), $ex->getMessage());
    }
```

```php
public function loginUser($credentials)
{
    $token = Auth::guard('user-api')->attempt($credentials);
    if (!$token) {
        return $this->returnError('E001', 'بيانات الدخول غير صحيحة');
    }
    $user = Auth::guard('user-api')->user();
    $user->api_token = $token;
    //return token
    return $this->returnData('data',new personResource($user), "data have returned");
}
```

Figure 13 login

2-register

```php
public function switchLogin(Request $request)
{...
}
public function switchRegister(Request $request)
{
    try {
        $role = $request->role;

        switch ($role) {

            case 'user':
                return $this->registerUser($request);
                break;
            case 'planner':
                return $this->registerPlanner($request);
                break;
            case 'hallowner':
                return $this->registerOwner($request);
                break;
                case 'supplier':
                    return $this->registerSupplier($request);
                    break;
            default:
                return response()->json([
                    'message' => 'ERORR',

                ], 201);;
                break;
        }
    } catch (\Exception $ex) {
        return $this->returnError($ex->getCode(), $ex->getMessage());
    }
}
```

```php
public function registerUser(Request $request)
{
    $validator = Validator::make($request->all(), [
        'name' => 'required|string|between:2,100',
        'email' => 'required|string|email|max:100|unique:admins|unique:users|unique:owners|unique:planners|unique:suppliers',
        'password' => 'required|string|min:6',
        'country' => 'required|string|max:100',
        'religion' => 'required|string|max:100',
        'gender' => 'required|string|max:100',
        'phone' => 'required|string|min:10|max:25',
        'photo',

    ]);
    if ($validator->fails()) {
        return response()->json($validator->errors()->toJson(), 400);
    }
    $user = User::create(array_merge(
        $validator->validated(),
        [
            'password' => bcrypt($request->password),
            'photo' => $this->uploadFile($request, 'usersImages', 'photo'),
        ]
    ));

    $credentials = $request->only(['email', 'password']);
    $token = Auth::guard('user-api')->attempt($credentials);
    if (!$token) {
        return $this->returnError('E001', 'بيانات الدخول غير صحيحة');
    }
    $userToken = Auth::guard('user-api')->user();
    $userToken->api_token = $token;

    return response()->json([
        'message' => 'User successfully registered',
        'data' => new personResource($userToken),
    ], 201);
}
    Omar, 4 weeks ago • first commit
```

Figure 14 register

3-profile

```php
public function anyProfile($email)
{

    $admin = Admin::where('email', $email)->first();
    $user = User::where('email', $email)->first();
    $owner = Owner::where('email', $email)->first();
    $planner = Planner::where('email', $email)->first();
    $supplier = Supplier::where('email', $email)->first();

    if ($admin) {
        return new adminResource($admin);
    } elseif($user){
        return new personResource($user);
    }
     elseif($owner){
        return new ownerResource($owner);
    }elseif($planner){
        return new plannersResource($planner);
    }elseif($supplier){
        return new supplierResource($supplier);
    } else {
        return response()->json(['error' => 'user not found'], 404);
    }
}
```

Figure 15 profile

4-BookRoom

```php
public function bookRoom(Request $request)
{
    // validate input data
    $validatedData = $request->validate([

        'hall_id'=> 'required|integer',
        'check_in_date'=> 'required|date',
        'check_out_date'=> 'required|date|after:check_in_date',
    ]);

    try {
        $the_user_id = Auth::guard('user-api')->user()->id;
    } catch (\Exception  $e) {
        return response()->json(['message' => 'Invalid token'], 401);
    }

    $hall = Hall::find($request->input('hall_id'));


    if (!$hall) {
        return response()->json(['error' => 'Hall not found'], 404);
    }else{
        if( $this->availablity_for_booking($request)== 0){
                $special_offer = $hall->offers()
                ->where('start_date', '<=', $request->input('check_in_date'))
                ->where('end_date', '>=', $request->input('check_out_date'))
                ->first();
                $price = $special_offer ? $special_offer->price : $hall->price;
                // create new booking record
                $booking = Booking::create([
                    'user_id' => auth::guard('user-api')->user()->id,
                    'user_name' => auth::guard('user-api')->user()->name,
                    'hall_id' => $validatedData['hall_id'],
                    'hall_name'=> $hall->name,
                    'check_in_date' => $validatedData['check_in_date'],
                    'check_out_date' => $validatedData['check_out_date'],
                    'price' => $price,
                ]);
                // return JSON response
                return response()->json([
                    'message' => 'Booking created successfully',
                    'booking' => $booking,
                ]);
            }
            else return response()->json([
                'message' => 'Hall Is Not AVILABLE ']);
        }
```

Figure 16 BookRoom

5-BookPlan

```php
public function bookPlan(Request $request)
{
    // validate input data
    $validatedData = $request->validate([
        'plan_id'=> 'integer',
        'check_in_date'=> 'required|date',
        'check_out_date'=> 'required|date|after:check_in_date',
    ]);
    try {
        $the_user_id = Auth::guard('user-api')->user()->id;
    } catch (\Exception  $e) {
        return response()->json(['message' => 'Invalid token'], 401);
    }

    $plan = Plan::find($request->input('plan_id'));
    $planname = $plan->plan_name;

    $planner = Planner::findOrFail($plan->planner_id);
    $planner_name = $planner->name;
    if (!$plan) {
        return response()->json(['error' => 'Hall not found'], 404);
    }else{
        // create new Planbooking record
        $bookingPlan = PlanRequest::create([
            'planner_id' => $plan->planner_id,
            'planner_name' => $planner_name,
            'user_id' => auth::guard('user-api')->user()->id,
            'user_name' => auth::guard('user-api')->user()->name,
            'plan_id' => $validatedData['plan_id'],
            'plan_name' =>$plan->name,
            'check_in_date' => $validatedData['check_in_date'],
            'check_out_date' => $validatedData['check_out_date'],
            'price' => $plan->price,
            'status' =>'unconfirmed',
        ]);
        // return JSON response
        return response()->json([
            'message' => 'Booking Plan created successfully',
            'booking' => $bookingPlan,
        ]);


    }

}
```

Figure 17 BookPlan

6-BookService

```php
public function bookSubService(Request $request)
{
    // validate input data
    $validatedData = $request->validate([


        'sub_id'=> 'integer',
        'check_in_date'=> 'required|date',
        'check_out_date'=> 'required|date|after:check_in_date',

    ]);

    try {
        $the_user_id = Auth::guard('user-api')->user()->id;
    } catch (\Exception  $e) {
        return response()->json(['message' => 'Invalid token'], 401);
    }

    $subservice = SubService::find($request->input('sub_id'));
    $subname = $subservice->sub_name;

    $supplier = Supplier::findOrFail($subservice->supplier_id);
    $supplier_name = $supplier->name;
    if (!$subservice) {
        return response()->json(['error' => 'Hall not found'], 404);
    }else{

        // create new Planbooking record
        $bookingSubservice = SubRequest::create([
            'supplier_id' => $subservice->supplier_id,
            'supplier_name' => $supplier_name,
            'user_id' => auth::guard('user-api')->user()->id,
            'user_name' => auth::guard('user-api')->user()->name,
            'sub_id' => $validatedData['sub_id'],
            'sub_name' =>$subservice->name,
            'check_in_date' => $validatedData['check_in_date'],
            'check_out_date' => $validatedData['check_out_date'],
            'price' => $subservice->price,
            'status' =>'unconfirmed',
        ]);
        // return JSON responses
        return response()->json([
            'message' => 'Booking SubService created successfully',
            'booking' => $bookingSubservice,
        ]);
        }       Omar, 3 weeks ago • supp;ier booking & anyprofile

}
```

Figure 18 BookService

7-addHall

```php
public function addHallRequests(Request $request)
{
    $validator = Validator::make($request->all(), [
        'name' => 'required|max:255',
        'address' => 'required|max:255',
        'country' => 'required|max:255',
        'city' => 'required|max:255',
        'street' => 'required|max:255',
        'rooms' => 'required',
        'chairs' => 'required',
        'price' => 'required',
        'hours' => 'required',
        'tables' => 'required',
        'type' => 'required|max:255',
        'capacity' => 'required',
        'available' => 'required',
        // 'start_party' => 'required',
        // 'end_party' => 'required',
    ]);
    if ($validator->fails()) {
        return $this->response(null, $validator->errors(), 400);
    }
                ///$the_owner_id = Auth::guard('owner-api')->user()->id;
    try {
        $the_owner_id = Auth::guard('owner-api')->user()->id;
    } catch (\Exception  $e) {
        return response()->json(['message' => 'Invalid token'], 401);
    }
    try {
        DB::beginTransaction();
        $result = Hall::create([
            'name' => $request->name,
            'address' => $request->address, 'country' => $request->country,
            'city' => $request->city,
            'street' => $request->street,
            'rooms' => $request->rooms,
            'chairs' => $request->chairs, 'price' => $request->price,
            'hours' => $request->hours, 'tables' => $request->tables,
            'type' => $request->type, 'capacity' => $request->capacity,
            'available' => $request->available,
            'verified' => "unconfirmed",

            'owner_id' => auth::guard('owner-api')->user()->id,

            'start_party' => $request->start_party,
            'end_party' => $request->end_party
        ]);
```

```php
        if (($request->photos)) {
            if($request->photos[0]){
            for ($i = 0; $i < count($request->photos); $i++) {
                $path = $this->uploadMultiFile($request, $i, 'hallPhotos', 'photos');
                Photo::create([
                    'photoname' => $path,
                    'hall_id' => $result->id,
                ]);
            }
        }}
        if ($request->videos) {
            if($request->videos[0]){
            for ($i = 0; $i < count($request->videos); $i++) {
                $path = $this->uploadMultiFile($request, $i, 'hallVideos', 'videos');
                Video::create([
                    'videoname' => $path,
                    'hall_id' => $result->id,
                ]);
            }
        }}
        if ($request->services) {
            $request->services[0];
            for ($i = 0; $i < count($request->services); $i++) {
                $services = $request->services;
                Service::create([
                    'servicename' => $services[$i],
                    'hall_id' => $result->id,
                ]);
            }
        }
        if ($request->shows) {
            $request->shows[0];
            for ($i = 0; $i < count($request->shows); $i++) {
                $shows = $request->shows;
                Show::create([
                    'showname' => $shows[$i],
                    'hall_id' => $result->id,
                ]);
            }
        }
        DB::commit();
        if ($result) {
            return $this->response($this->hallResources($result), 'done', 201);
        } else {
            return $this->response(null, 'halls is not saved', 405);
        }
    } catch (Exception $e) {      You, 4 weeks ago • first commit
        DB::rollback();
```

8-confirmAndRejectBooking

```php
    public function confirmBooking($bookingId){
        $booking = Booking::findOrFail($bookingId);

        $booking->status = 'confirmed';
        $booking->save();

        return response()->json([
            'message' => 'Booking confirmed successfully',
            'data' => $booking
        ], 200);    }

    public function rejectBooking($bookingId){
        $booking = Booking::findOrFail($bookingId);

        $booking->status = 'cancelled';
        $booking->save();

        return response()->json([
            'message' => 'Booking cancelled successfully',
            'data' => $booking
        ], 200);    }
```

Figure 19 confirmAndRejectBooking

9-addOffer

```php
public function store(Request $request)
{       Omar, 4 weeks ago • first commit
    $user = auth()->guard('admin-api')->user();
    if (!$user) {
        return response()->json(['error' => 'Unauthorized'], 401);
    }
    else{
    $hall = Hall::find($request->input('hall_id'));

    if (!$hall) {
        return response()->json(['error' => 'Hall not found'], 404);
    }

    $special_offer = new Offer([
        'hall_id' => $request->input('hall_id'),
        'hall_name' => $request->input('hall_name'),
        'package_description' => $request->input('package_description'),
        'start_date' => $request->input('start_date'),
        'end_date' => $request->input('end_date'),
        'price' => $request->input('price'),
    ]);

    $hall->offers()->save($special_offer);

    return response()->json([

        'message' => 'Special offer created',
        'offer' =>$special_offer



], 201);
    }
}
```

10-addComment

```php
public function addComment(Request $request,  $hall_id)
{
    $thehall = Hall::where('id',$hall_id)->first();

    if($thehall)/* Check if Hall Id is valid */{


        $rules = [
            'content' => 'required|string',
        ];
        $validator = Validator::make($request->all(),$rules);
        if ($validator->fails())/* Check if Comment's content is valid */ {
            $code = $this->returnCodeAccordingToInput($validator);
            return $this->returnValidationError($code, $validator);
        }
        else{
        $comment = Comment :: create ([
            'hall_id'=>$thehall->id,
            'user_id'=> auth()->user()->id,
            'content'=>$request->content,
        ]);
        $comment->load('users');
        $comment->load('admins');
        $comment->load('users');
        $comment->load('halls');
        return response()->json(['message' => 'Comment added.']);}
    }
    else{
        response()->json(['message' => 'HALL ID IS WRONG.']);
    }


}
```

Figure 20 addComment

11-addLike

```php
public function addLike(Request $request,  $hall_id)
{

    $thehall = Hall::where('id',$hall_id)->first();

    if($thehall)/* Check if Hall Id is valid */{
            $like = Like :: create ([
                'hall_id'=>$thehall->id,
                'user_id'=>  auth()->user()->id,
            ]);
            $like->load('users');
            $like->load('halls');
    return response()->json(['message' => 'Post liked.']);
    }
    else{
        return $this->removeLike($hall_id);
    }


}
```

12-guards

Figure 21 guards

13-providers

Figure 22 providers

14-cors

```php
    'paths' => ['*'],

    'allowed_methods' => ['*'],

    'allowed_origins' => ['*'],

    'allowed_origins_patterns' => [],

    'allowed_headers' => ['*','Content-Type','Authorization','auth-token'],

    'exposed_headers' => [],


    'max_age' => 0,

    'supports_credentials'=>false,

];
```

Figure 23 cors

15-verifyToken

```php
class VerifyToken
{
    public function handle($request, Closure $next, $guard = null)
    {
        if (!$request->bearerToken()) {
            throw new AuthenticationException('Unauthorized');
        }

        if (auth($guard)->check()) {
            return $next($request);
        }

        throw new AuthenticationException('Unauthorized');
    }
}
```

Figure 24 verifyToken

16-assignGuard

```php
public function handle($request, Closure $next, $guard = null)
{
    if($guard != null){
        auth()->shouldUse($guard); //shoud you user guard / table
        $token = $request->header('auth-token');
        $request->headers->set('auth-token', (string) $token, true);
        $request->headers->set('Authorization', 'Bearer '.$token, true);
        try {
            //  $user = $this->auth->authenticate($request);  //check authenticted user
            $user = JWTAuth::parseToken()->authenticate();
        } catch (TokenExpiredException $e) {
            return  $this -> returnError('401','Unauthenticated user');
        } catch (JWTException $e) {

            return  $this -> returnError('', 'token_invalid'.$e->getMessage());
        }

    }
    return $next($request);
}
```

Figure 25 assignGuard

Chapter 5

Testing.

In this chapter we tested our application and tried different test cases to ensure that the application have reached the level of quality we are targeting

First, there are two fundamental purposes of testing:

- First, **testing is about verifying that what was specified is what was delivered:** it

verifies that the product (system) meets the functional, performance, design, and

implementation requirements identified in the procurement specifications.

- Second, **testing is about managing risk.** The testing program is used to identify.

when the work has been “completed” so that the contract can be closed, the vendor paid, and the system shifted by the agency into the warranty and

maintenance phase of the project.

There are 2 types of testing (Functional and non-Functional testing)

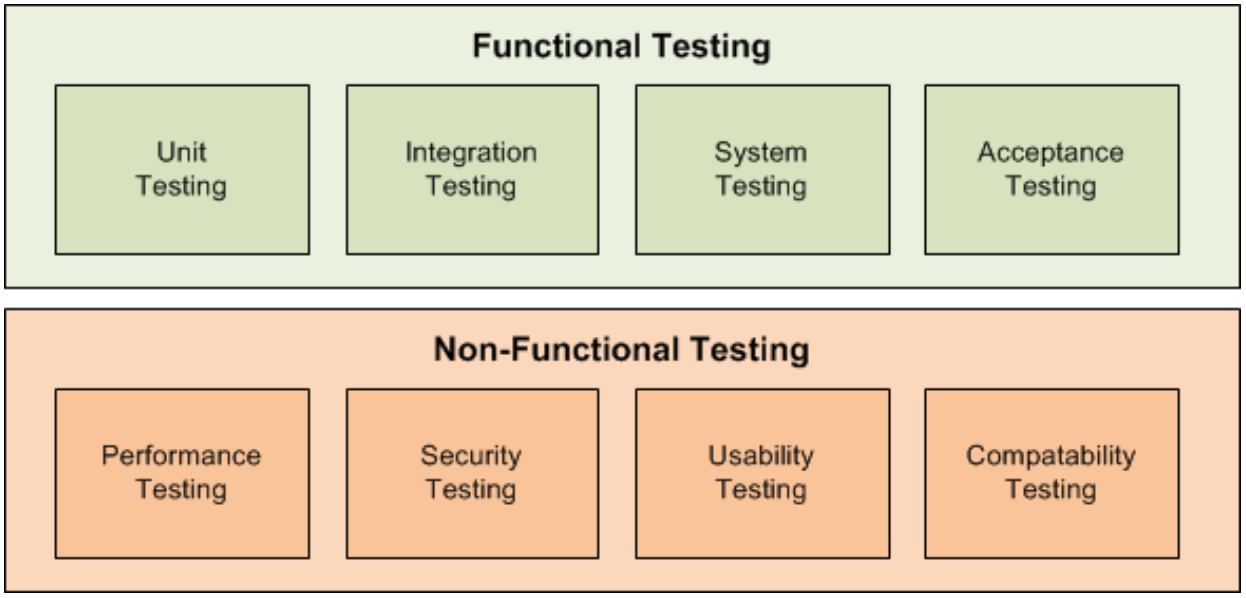


Figure 26 Functional and non-Functional testing

5.1 Functional Testing

5.1.1 Unit Testing

Testing of individual items (e.g., modules, programs, objects, classes, etc.)

As part of the coding phase, in isolation from other development items and the system.

5.1.2 Integration Testing

Testing the interfaces between major (e.g. systems level application modules) and minor (e.g. individual programs or components) items within the application, also, testing the application as whole after committing a file into a branch on GitHub

5.1.3 System Testing

Testing a system behaviour after it is fully integrated, when development is finished, hence, the system can be tested as complete entity.

5.1.4 System Testing

Testing a system behaviour after it is fully integrated, when development is finished, hence, the system can be tested as complete entity.

5.1.5 Regression Testing

To check older functionality after integrating new functionality.

5.1.6 Acceptance testing

To check older functionality after integrating new functionality.

5.2 Non-Functional Testing

5.2.1 Performance Testing

Accomplished a designated function regarding processing time and through put rate.

5.2.2 Load Testing

Measuring the behaviour of within creasing load that can be handled by the component or system.

5.2.3 Stress Testing

Evaluate a system or component at or beyond the limits of its specified requirements.

5.2.4 Security Testing

Testing how well the system protects against unauthorized internal or external access.

5.3 Unit Testing Results:

Registration Test case:

DEFECTS	EXPECTED RESULTS
Entered an empty value in First Name Field	Error Message appeared as first name is a required field
Entered an empty value in Last Name Field	Error Message appeared as last name is a required field
Entered an existing E-mail	Error Message appeared as e-mail is already taken, please enter a unique username
Entered an E-mail without "@"	Error Message appeared as e-mail should contains a "@" symbol
Entered a Password with length less than 10	Error Message appeared as password's length should at least 10
Entered a Password without an uppercase letter	Error Message appeared as password should contain uppercase letter
Entered a Password without a lowercase letter	Error Message appeared as password should contain lowercase letter
Entered a Password without a special character (@, _, -, #, *, &, ~, etc..)	Error Message appeared as password should contain a special character
Entered a Password without digit numbers	Error Message appeared as password should contain a digit number

Table 1 Registration Test case

Note: By entering valid values, the registration process will be done. And the patient will be added to the database.

Login Test case:

DEFECTS	EXPECTED RESULTS
Not entering the email	Error Message appeared as email is a required field
Not entering the Password	Error Message appeared as Password is a required field
Entering Invalid email	Error Message appeared as This email is Invalid
Entering Invalid Password	Error Message appeared as This Password is incorrect

Table 2 Login Test case

Note: By entering valid values, the login process will be done. And the user will be directed to the home page.

Booking (Book Hall, Book Plan) Test case:

DEFECTS	EXPECTED RESULTS
Not selected item	Error Message appeared as (item) is a required field
Selecting a Reserved item	Error Message appeared this (item) is already reserved by another patient
Not selected Start Time or End Time	Error Message appeared as start time and end time are required fields
Selecting End Time less than Start Time	Error Message appeared as end time shouldn't be less than start time

Table 3 Booking

Note: By entering valid values, the reservation process will be done.

Hall Owner (Add Hall) Test case:

DEFECTS	EXPECTED RESULTS
Entered an empty value in Name Field	Error Message appeared as name is a required field
Entered an empty value in Address Field	Error Message appeared as address is a required field
Entered an empty value in capacity Field	Error Message appeared as **capacity** is a required field
Entered an empty value in type Field	Error Message appeared as **type** is a required field
Entered an empty value in price Field	Error Message appeared as **price** is a required field
Entered an empty value in chairs Field	Error Message appeared as **chairs** is a required field
Entered an empty value in tables Field	Error Message appeared as **tables** is a required field
Selecting a quantity for a Hall info that is either zero or Negative	Error Message appeared product quantity should not be zero or Negative

Table 4 Hall Owner

Note: By entering valid values, the Add Hall process will be done.

Planner/Supplier (Plan, Product) Test case:

DEFECTS	EXPECTED RESULTS
Entered an empty value in Name Field	Error Message appeared as name is a required field
Entered an empty value in Price Field	Error Message appeared as address is a required field
Entered an empty value in Description Field	Error Message appeared as **Description** is a required field
Entered an invalid price Field	Error Message appeared as **price** is invalid

Table 5 Planner/Supplier

Note: By entering valid values, the Addition process will be done.

Admin (Add Offers) Test case:

DEFECTS	EXPECTED RESULTS
Not selected a Hall	Error Message appeared as Hall is a required field
Entered an empty value in price Field	Error Message appeared as **price** is a required field
Not selected Start Time or End Time	Error Message appeared as start time and end time are required fields
Selecting End Time less than Start Time	Error Message appeared as end time shouldn't be less than start time
Selecting a quantity for a Hall info that is either zero or Negative	Error Message appeared product quantity should not be zero or Negative

Table 6 Admin

Note: By entering valid values, the add Offers process will be done.

Chapter 6

Discussion, Conclusions, and Future Work.

Discussion:

A wedding planner website can be a powerful tool for wedding planners to showcase their services, attract potential clients and manage their projects. When a wedding planner website is successful, there are several achievements that can be realized, including:

1- Attracting and Engaging Potential Clients: One of the primary goals of a wedding planner website is to attract and engage potential clients. A successful wedding planner website can do this by providing detailed information about the services offered, showcasing previous work, and making it easy for clients to contact the wedding planner…. And that what we do by design user interface attractive.

2- Streamlining the Planning Process: A wedding planner website can help streamline the planning process by providing tools and resources to help clients plan their weddings more efficiently. This can include features such as budget calculators, vendor directories, and planning checklists…. And that what we do by make the website easy to use and easy to book a hall and not feel confused when you enter the website.

3- Building Trust and Credibility: A successful wedding planner website is able to build trust and credibility with potential clients by providing testimonials from previous clients, showcasing awards and certifications, and providing a professional and polished appearance…. And we do that by provide all information about all halls and wedding planner that make user feel safe.

4- Managing Projects: A wedding planner website can also help manage projects by providing tools for tracking progress, communicating with clients, and managing deadlines. This can help ensure that projects are completed on time and within budget….And we do that by provide a way for connect with user and the owner of hall.

5- Attracting and Engaging Potential Clients: A successful wedding planner website should have a clear and concise message that communicates the

benefits of working with the wedding planner. It should be visually appealing and easy to navigate, with a focus on providing valuable information and resources to potential clients. This can include features such as a blog, a portfolio of previous work, and a contact form for inquiries…. And we do that by provide some offer that you can add to the hall that you choose.

6- Building Trust and Credibility: For potential clients to trust a wedding planner, they need to feel confident in their skills and experience. A successful wedding planner website should showcase the wedding planner's credentials, such as awards, certifications, and testimonials from previous clients. This can help build trust and credibility with potential clients and differentiate the wedding planner from competitors…. And we achieved that by show the previous booking for each hall.

7- Managing Projects: A wedding planner website can also help manage projects by providing tools for tracking progress, communicating with clients, and managing deadlines. For example, the website might include a client portal that allows clients to view their project timeline, budget, and vendor information. This can help ensure that projects are completed on time and within budget and can also improve communication and collaboration between the wedding planner and the client…And that we do by provide page for contact us have email and name and number.

- We faced a difficulty in the api and force a lot of error when we try to attach the work between back-end and front-end.

-we faced a difficulty in delay delivery work from some member.

-we found difficulty in learning some new tools that we used in project.

- we found a lot of error that take a lot of time to solve it.

- we late to deliver task to each other because of exams that we have to end it in collage.

-we faced a difficulty because there no cooperation between team.

Conclusion:

In a project wedding planner, the overall goal is to successfully plan and execute a wedding event that meets the client's expectations and creates a memorable experience for all involved. Here are some specific achievements that can be realized in a project wedding planner:

1- Meeting the Client's Vision: One of the primary goals of a wedding planner is to bring the client's vision to life. This involves understanding the client's preferences, style, and expectations and incorporating these into the planning process to create a wedding that truly reflects the client's personality and desires.

2- Managing the Budget: A successful wedding planner can create a budget that works for the client's needs and then manage the expenses throughout the planning process to ensure that the wedding stays within budget.

3- Coordinating Vendors: A wedding involves many different vendors, from the caterers to the florists to the musicians. A successful wedding planner is able to coordinate all of these vendors, ensuring that they arrive on time and that everything runs smoothly on the day of the wedding.

4- Creating a Memorable Experience: Ultimately, the goal of a wedding planner is to create a memorable experience for the client and their guests. A successful wedding planner can create a wedding that is not only beautiful and well-organized but also reflects the client's personality and leaves a lasting impression on everyone who attends.

Future Work

-One of the ideas we had but there wasn't a plenty of time to add it in our project that we could put many languages on the website.

- One of the ideas we had but there wasn't a plenty of time to add it in our project that we add Masary ,fawry and aman system.

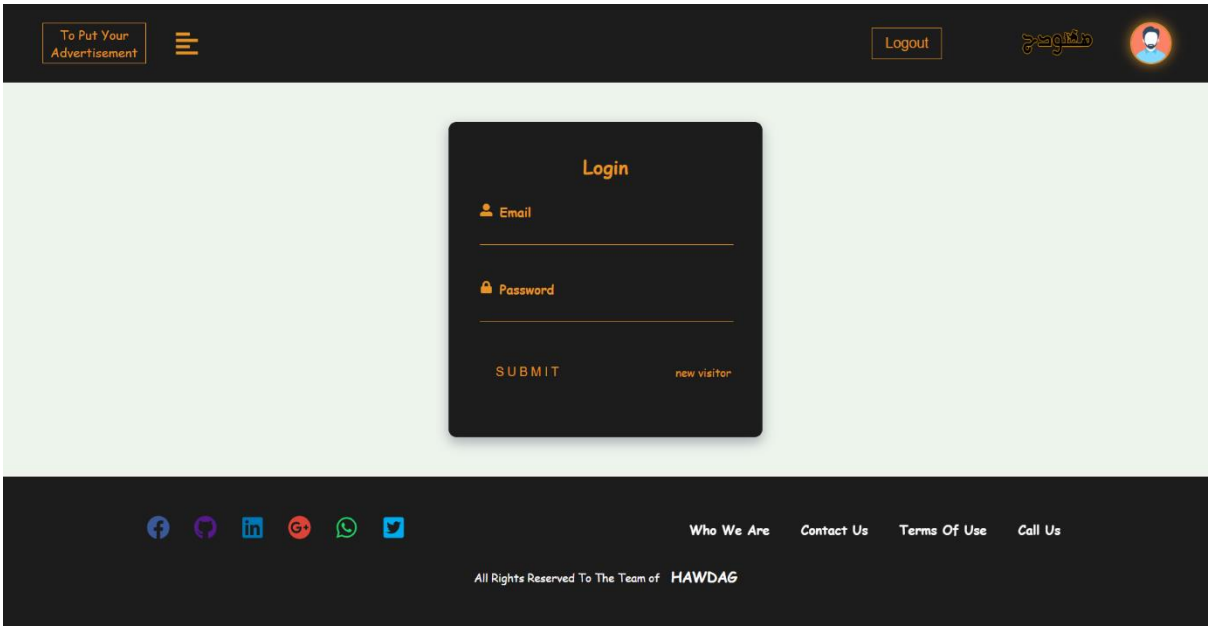
- User will be able to create his own package and choose whatever he wants to have in his wedding.

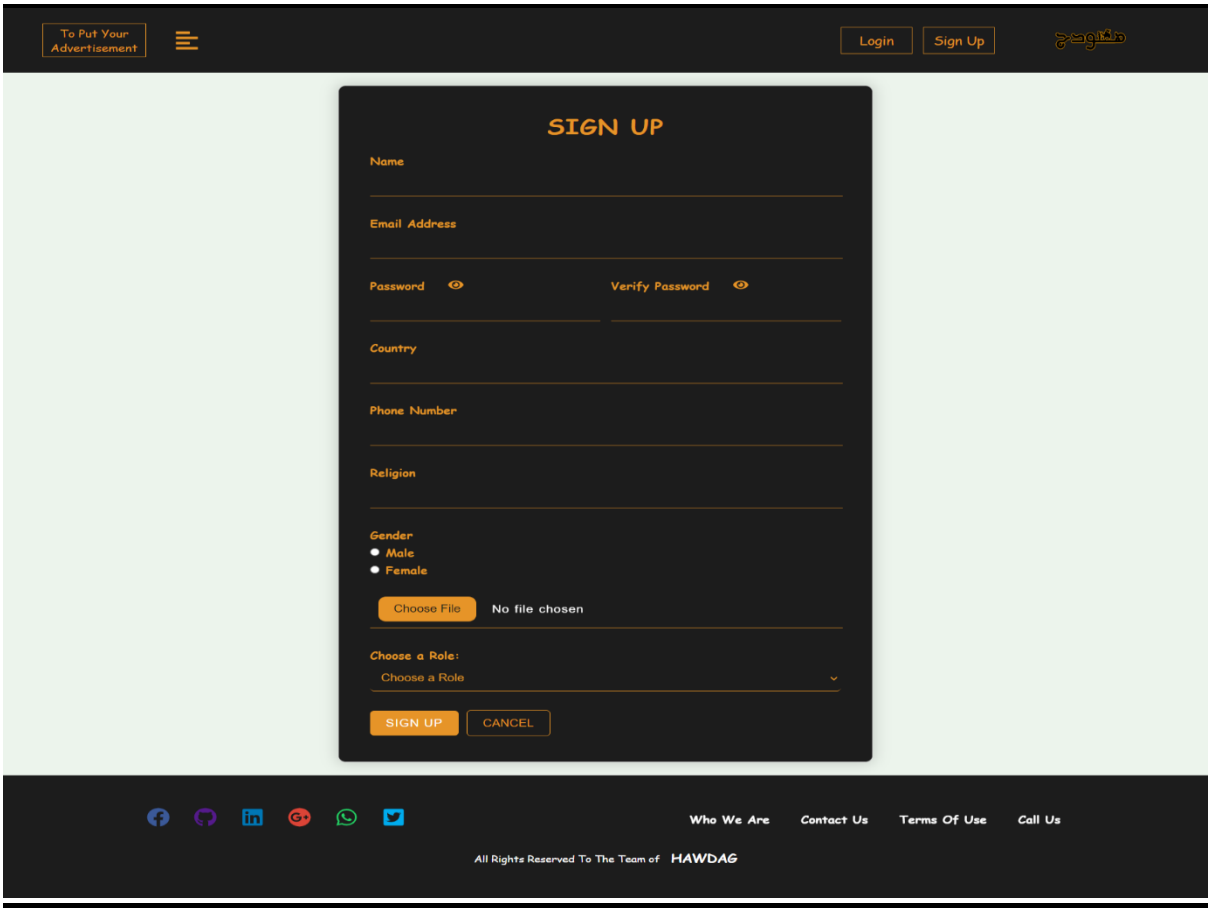
- Hawdag will be integrated with google maps to give the client more convenient experience to reach his halls.

- Hawdag will have its own chat to enable all it's users to contact each other.

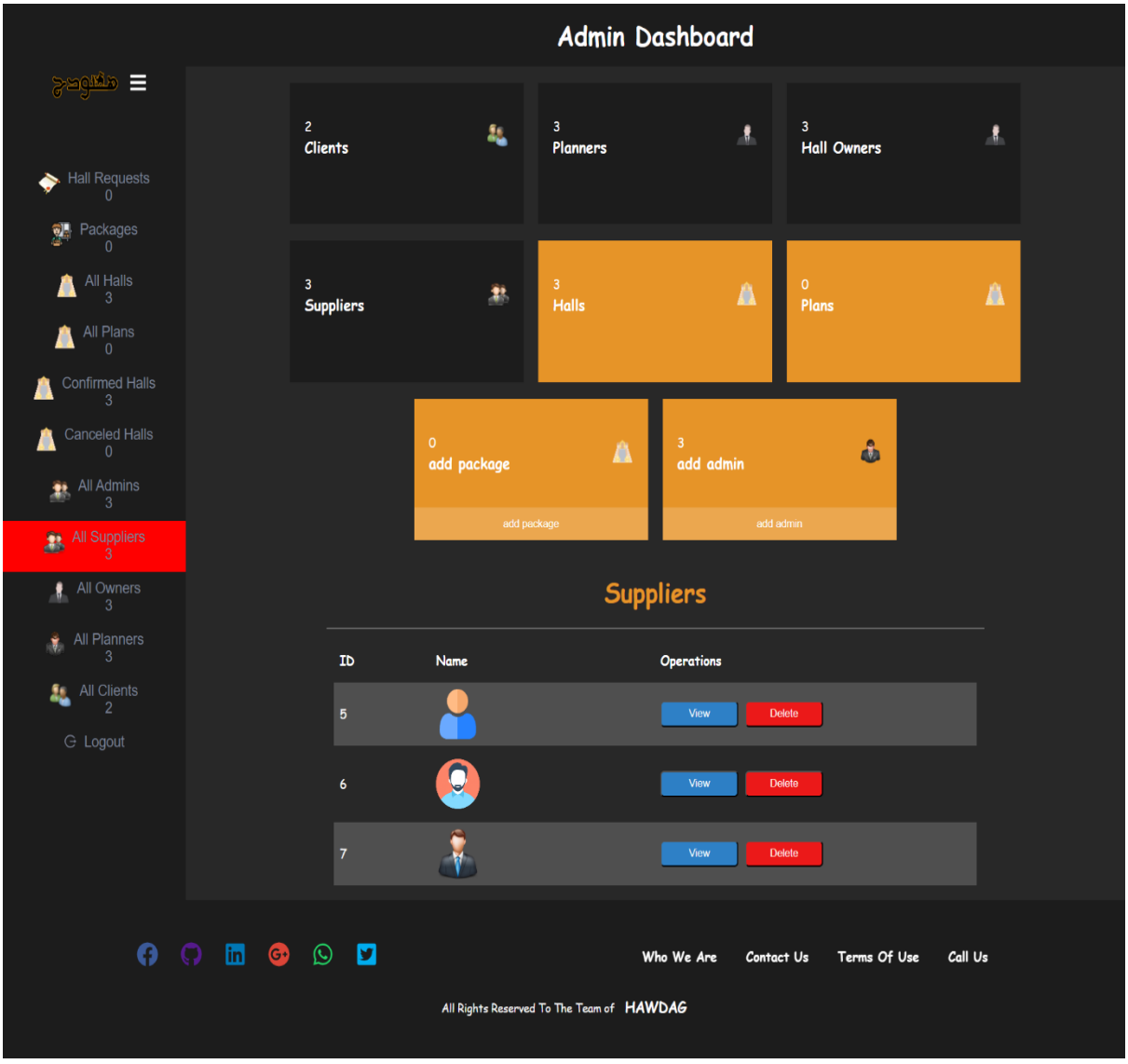
Some pages from our website


To Put Your Advertisement
Login
Sign Up
Discover exquisite wedding halls that will make your special day unforgettable.
How to Plan a Wedding
Our Recommendation For You
Mobile Number
1(234)567-891
Location
Giza-Egypt
Email For Contact
hawdag@email.com
Contact Us
Name
Write Your Message
SEND
Who Are We
We are a group of young people who aim to make it easier for users to find the most suitable halls for their events
Who We Are
Contact Us
Terms Of Use
Call Us
All Rights Reserved To The Team of HAWDAG


To Put Your Advertisement
Logout
Login
Email
Password
SUBMIT
new visitor
Who We Are
Contact Us
Terms Of Use
Call Us
All Rights Reserved To The Team of HAWDAG


To Put Your Advertisement
Login
Sign Up
SIGN UP
Name
Email Address
Password
Verify Password
Country
Phone Number
Religion
Gender
Male
Female
Choose File
No file chosen
Choose a Role:
Choose a Role
SIGN UP
CANCEL
Who We Are
Contact Us
Terms Of Use
Call Us
All Rights Reserved To The Team of HAWDAG


Services
Dj
a wonderfull dj making the wedding so funny and exciting and play a lot of nice songs
500
Details
Edit
cake
a delisious cake with a good teste
1000
Details
Edit
To Put Your Advertisement
Login
Sign Up
Wedding Planners
Planner1
Details
Planner3
Details
Planner5
Details
Hall Owners
Owner1
Details
Owner3
Details
Owner5
Details
Suppliers
supplier
Details
Supplier2
Details
Supplier3
Details
Who We Are
Contact Us
Terms Of Use
Call Us
All Rights Reserved To The Team of HAWDAG


Admin Dashboard
Hall Requests 0
Packages 0
All Halls 3
All Plans 0
Confirmed Halls 3
Canceled Halls 0
All Admins 3
All Suppliers 3
All Owners 3
All Planners 3
All Clients 2
Logout
2 Clients
3 Planners
3 Hall Owners
3 Suppliers
3 Halls
0 Plans
0 add package
add package
3 add admin
add admin
Suppliers
ID
Name
Operations
5
View
Delete
6
View
Delete
7
View
Delete
Who We Are
Contact Us
Terms Of Use
Call Us
All Rights Reserved To The Team of HAWDAG