Software Requirements Specification

for

Social HUB

A project to bring people together

Prepared by **Zeynep Deniz Yılmaz**
Zeynep Ece Sarıoğlu
Zeynep Gültekin
Sarp Başaraner

zeyneplervesarp

09.04.2022

Table of Contents

Introduction

Purpose

The purpose of SocialHub is to create an ecosystem where only time is used as a currency. It is a virtual environment that brings people together online and offline, enables them to share their experiences, and learn new things, make friends and expand their horizons.
This document covers the SocialHub project and all of its features, use cases, scenarios, UML and sequence diagrams.

Document Conventions

The overall touch and feel of the project is done with Vue Argon System created by Creative Tim. The reference can be found in the references section. The requirements mentioned below state a higher-level description of the system. The detailing is done through sequence diagrams and issues.

Intended Audience and Reading Suggestions

This document is intended for the customer, Product Owner from the client-side, as well as the developers who need to build and test the application.

Project Scope

SocialHub is a web application that serves as an open platform for internet users. The system is designed to create an environment where users can offer services based on their expertise and also participate in others' services.

More specifically, the platform is created to register users, get and display their profile information, display their balance, list services and filter and sort them, display services, and get notifications for various events. There is also administration activity that enables the administrative users to handle member created content, handle users, and feature services for community fostering.

SocialHub facilitates communication between its users as it offers real time events called services. The system contains a relational database containing information on users, services, tags, notifications, user followings and approvals.

References

This project has used a free design template by Creative Tim, which can be found here:
https://www.creative-tim.com/product/vue-argon-design-system

Overall Description

Product Features

The system will be open to the internet, and unregistered users will be able to display a sample of services with limited information. Users will be able to display all services, create services, receive and display notifications, request services, approve and deny requests to their services, follow and be followed by other users, display the profile of other users and approve services.
Users will be the primary users and unregistered users will be the secondary users.

User Documentation

The user documentation will be provided at a later time.

Assumptions and Dependencies

SocialHub is deployed on an AWS server. The capacity of the server can be increased or decreased by demand. The system can be run on other service providers if needed. The software developed assumes the use of Docker for containerizing the application. The speed of the system will depend on the server capacity.

1. System Features (Functional Requirements)

In this section, the functional requirements are categorized by features.

1.1. Services

1.1.1. Creating Services

1.1.1.1. The system shall enable users to create services.
1.1.1.2. The system shall enable users to apply to services.
1.1.1.3. The system shall link only one user as a creator of a service.
1.1.1.4. The system shall allow only registered users to create services.
1.1.1.5. The system shall enable users to apply for other users' services.
1.1.1.6. The system shall maintain a quota for attendees of a service.
1.1.1.7. The system shall maintain date information of a service.
1.1.1.8. The system shall maintain time information of a service.
1.1.1.9. The system shall maintain description information of a service.
1.1.1.10. The system shall maintain "application deadline" information of a service. *
1.1.1.11. The system shall maintain location information of a service.
1.1.1.12. The system shall enable users to choose between a physical service or a virtual service while creating a service. *

1.1.1.13. The system shall maintain zoom link information of a virtual service *

1.1.2. Applying for Services

1.1.2.1. The system shall allow users with enough time credits to apply for a service.

1.1.3. Listing Services

1.1.3.1. The system shall display services sorted by the distance to a logged-in user's location.
1.1.3.2. The system shall display services sorted by their created date and service's date.
1.1.3.3. The system shall alow users to search for services by entering location.

1.1.4 Editing Services

1.1.4.1. The system shall notify the applicant users when the creator changes something in the service.
1.1.4.2. The system shall allow admins to edit services.
1.1.4.3. The system shall allow admins to edit events.
1.1.4.4. The system shall allow admins to delete services.
1.1.4.5. The system shall allow admins to delete events.

1.1.5. Deadlines of Services

1.1.5.1. The system shall allow users to apply for a virtual service until 30 minutes before the start time. *
1.1.5.2. The system shall allow users to edit the service until 24 hours is left to service start time. *
1.1.5.3. The system shall allow users to apply for a service before the service deadline time. *

1.1.6. The Handshake of Services

1.1.6.1. The system shall enable the attending users to approve that they attended the service.
1.1.6.2. The system shall enable the creator user to approve that the service is completed.
1.1.6.3. The system shall allow a user to complete a service when all attendees have confirmed that the service took place.

1.2. Events

1.2.1. Creating Events

1.2.1.1. The system shall enable users to create events.
1.2.1.2. The system shall set the time credits to attend an event zero when creating the event.
1.2.1.3. The system shall set the quota of an event to limitless when creating the event.
1.2.1.4. The system shall maintain the name of the event.
1.2.1.5. The system shall maintain the short description of the event.
1.2.1.6. The system shall maintain the selected tags of the event.
1.2.1.7. The system shall maintain the location of the event.

1.2.1.8. The system shall maintain the date of the event.

1.2.2. Attending Events

1.2.1.1. The system shall allow users to participate in events.
1.2.1.2. The system shall maintain the guests of an event.

1.3. Tags

1.3.1. Creating Tags

1.3.1.1. The system shall create tags if it does not already exist in the system.
1.3.1.2. The system shall allow user to enter any tag except illegal tags.

1.3.2. Adding tags

1.3.2.1. The system shall maintain tags linked to the users skills.
1.3.2.2. The system shall maintain tags linked to the users interests.
1.3.2.3. The system shall maintain tags linked to services.
1.3.2.4. The system shall maintain tags linked to events.

1.3.3. Handling Tags

1.3.3.1. The system shall maintain a list of illegal tags.

1.4. Notifications

1.4.1. Sending Notifications

1.4.1.1. The system shall send notification to a user when another user follows them.
1.4.1.2. The system shall send notification to a user when another user approves their application to a service.
1.4.1.3. The system shall send notification to a user when another user denies their application to a service.
1.4.1.4. The system shall send notification to a service creator user when another user applies to their service.
1.4.1.5. The system shall send notification to a participating user when the service creator edits their service.
1.4.1.5. The system shall send notification to a participating user when the service creator edits their event.

1.4.2. Handling Notifications

1.4.2.1. The system shall maintain the date of the notification.
1.4.2.1. The system shall maintain the link to the event belonging to the notification.
1.4.2.2. The system shall mark notifications as read when the receiving user reads them.

1.5. Time Credit Dynamics

1.5.1 Obtaining Time Credits

1.5.1.1. The system shall increase time credits of a creator user by specified credit amount of a service, when the service is completed.

1.5.2. Holding Time Credits

1.5.2.1. The system shall maintain credits on hold of a user.
1.5.2.2. The system shall hold credits of an applicant user, until the service they applied for is complete.

1.5.3. Losing Time Credits

1.5.3.1. The system shall decrease the attending user's time credits after the service the user attended is completed.

1.5.4. Handling Time Credits

1.5.4.1. The system shall maintain an upper limit of the user time credits.
1.5.4.2. The system shall maintain an lower limit of the user time credits.
1.5.4.3. The system shall maintain a time credit quota for every user.
1.5.4.4. The system shall assign 5 time credits to a newly registered user by default.
1.5.4.5. The system shall display the time credit to the owner user only.

1.5.5. Time Credits Interacting With Other Features

1.5.5.1. The system shall allow users to create new services unless they reach their upper time credit limit.
1.5.5.2. The system shall allow users to apply for services unless they reach their lower time credit limit.

1.6. Reputation Dynamics

1.6.1. Handling Reputation Points

1.6.1.1. The system shall maintain reputation points for every user.
1.6.1.2. The system shall assign 10 reputation points to the newly signed-up users by default.

1.6.2. Obtaining Reputation Points

1.6.2.1. The system shall enable users to rate the creator of the service during the completion of the service.
1.6.2.2. The system shall allow users to select the rating from 0 to 5.
1.6.2.3. The system shall calculate the reputation points from the ratings that are given to that user.

1.6.3. Losing Reputation Points

1.6.3.1. The system shall decrease 10 reputation points from the user that does not show up to the service they created.
1.6.3.2. The system shall decrease 5 reputation points from the creator user if the user cancells the service after the cancellation deadline.
1.6.3.3. The system shall decrease 5 reputation points from the applicant user if the applicant user cancels the application of the service after the cancellation deadline.
1.6.3.4. The system shall maintain a users reputation point to the lower limit of 0.

1.7. Administration

1.7.1. Admin Functionality

1.7.1.1. The system shall contain admin roles.
1.7.1.2. The admins can review users profiles.
1.7.1.3. The admins can flag users profiles.
1.7.1.4. The admins can review services.
1.7.1.5. The admins can flag services.
1.7.1.6. The admins can review events.
1.7.1.7. The admins can flag events.
1.7.1.8. The admins can remove services from the system.*
1.7.1.9. The admins can remove events from the system.*
1.7.1.10. The admins can remove users from the system.*
1.7.1.11. The admin shall pick 3 services to be featured. *
1.7.1.12. The system shall display featured services to the user. *

1.7.2. Admin Dashboard Functionality *

1.7.2.1. The system shall display to an admin when users are attending to a service.
1.7.2.2. The system shall display to an admin when the users are attending to an event.
1.7.2.3. The system shall display to an admin when the users are creating a service.
1.7.2.4. The system shall display to an admin when the users are creating an event.
1.7.2.5. The system shall display to an admin every successful user login.
1.7.2.6. The system shall display to an admin every unsuccessful user login.

1.8. Flagging

1.8.1. The system shall enable users to flag services.
1.8.2. The system shall enable the admin to review the service, when the service is flagged by 5 different users.
1.8.3. The system shall enable users to flag events as "not appropriate".
1.8.4. The system shall enable users to flag other users.

1.9. User Activity *

1.9.1. The system shall display the service attendance activity of the user to the user who follow them.
1.9.2. The system shall display the event attendance activity of the user to the user who follow them.
1.9.3. The system shall display the service creation activity of the user to the user who follow them.
1.9.4. The system shall display the event creation activity of the user to the user who follow them.
1.9.5. The system shall display the follow activity of the user to the user who follow them.

1.10. Search *

1.10.1 The system shall enable users to display all services.
1.10.2 The system shall enable users to filter all displayed services.
1.10.3. The system shall display personalized search results for each user.
1.10.4. The system shall personalize search results according to the follow list, tags, events they have attended to.

1.11. Activity Logging *

1.11.1. The System shall log activity when a user follow another user.
1.11.2. The System shall log activity when a user applies for a service.
1.11.3. The System shall log activity when a user attends an event.
1.11.4. The System shall log activity when a user is accepted to take a service.
1.11.5. The System shall log activity when a user logs in successfully.
1.11.6. The System shall log activity when a user attempts to log in unsuccessfully.

1.12. Recommendation *

1.12.1. The System shall display new users recommendations based on their tags.
1.12.2. The System shall display recommendations based on the users behaviour.
1.12.3. The system shall recommend to the users based on the past events that they have participated.

1.13. Badge Dynamics

1.13.1. The system shall display badges of a user to other users and guests.

1.13.2. The system shall assign a "newcomer" badge to users that are in the system for less than 3 months.
1.13.3. The system shall assign a "newcomer" badge to users that have participated in less than or equal to 10 services.
1.13.4. The system shall assign a "mentor" badge to a user that have participated in 10 services from a newcomer.
1.13.5. The system shall assign a "super mentor" badge to a user that have participated in 10 services from a newcomer.
1.13.6. The system shall assign a "regular" badge to a user that have participated in 20 services.
1.13.7. The system shall assign a "community builder" badge to a user that organized more than 10 events.
1.13.8. The system shall assign a "guru" badge to a user that have more than 50 reputation points.

2. Non-Functional Requirements

2.1. Usability

2.2. Security

2.2.1. Registration

2.2.1.1. The system shall allow guests to sign up to the system.
2.2.1.2. The system shall request from the guest username for sign up.
2.2.1.3. The system shall request from the guest password for sign up.
2.2.1.4. The system shall request from the guest location for sign up.
2.2.1.5. The system shall request from the guest small biography for sign up.
2.2.1.6. The system shall request from the guest to pick tags from a predefined list of tags for sign up.
2.2.1.7. The system shall display the location that the user have chosen on the map while registering.
2.2.1.8. The system shall display profile information to the user after authenticating to the system.
2.2.1.9. The system shall display detailed service information to the user after authenticating to the system.
2.2.1.10. The system shall display filtered and sorted service information after user authenticates to the system.

2.2.2. Authorization

2.2.2.1. The system shall generate a token for user authentication.
2.2.2.2. The system shall authenticate registered and logged-in users.
2.2.2.3. The system shall display an error when making an unauthorized request.

2.2.3. Users and Authorization

2.2.3.1 The system shall have member type users.
2.2.3.2. The system shall have admin type users.
2.2.3.3. The system shall allow users to display their own profiles.
2.2.3.4. The system shall allow users to view other users' profiles.
2.2.3.5. The system shall allow users to create new services.
2.2.3.6. The system shall allow users to join services.
2.2.3.7. The system shall display to the user the requests that are made to their services.
2.2.3.8. The system shall display notifications to the user.
2.2.3.9. The system shall allow users to follow other registered users.
2.2.3.10. The system shall enable users to be followed other registered users.
2.2.3.11. The system shall enable users to flag other services *
2.2.3.12. The system shall allow guests to only view services.

3. External Interface Requirements

3.1. User Interfaces

3.1.1. The services page shall display a map with a point on the location of the service.
3.1.2. The system shall display a notification window accessible from the navigation bar.
3.1.3. The system shall enable users to view all notifications they have received.
3.1.4. The system shall display the unread notifications marked with bold text.
3.1.5. The system shall display read notifications marked with gray, faint text.
3.1.6. The system shall display to the user their activity history on their profile page.

3.1.1. Homepage and Feed

3.1.1.1. The system shall display Homepage with a link to user page.
3.1.1.2. The system shall display the home page after user logs in.
3.1.1.3. The system shall display to the user services created by the users that they follow. *
3.1.1.4. The system shall display to the user events created by the users that they follow. *
3.1.1.5. The system shall display a "featured services" section to the users homepage.
3.1.1.6. The system shall display a list of services in users homepage.
3.1.1.7. The system shall display a list of events in users homepage.
3.1.1.8. The system shall display a new tag page for members from the homepage.
3.1.1.9. The system shall display time credits on the user's own profile page.

3.2. System Requirements

3.2.1. The system shall work as a web application.
3.2.2. The systems backend shall function as a REST API.
3.2.3. The systems frontend shall function as a web application.
3.2.4. The systems frontend shall consume the backend API.
3.2.5. The system shall be accessible on web.
3.2.6. The system shall be deployed on cloud.
3.2.7. Geolocation shall be used for the location information.

Appendix A: Glossary

These are some terms that require description in order for the requirements to be more understandable.

administrators: people who remove sensitive or reported content from the platform to keep it safe and in line with the community rules. They are hired by the management of the platform.

balance: the currency for this platform. (For example, if a service has lasted an hour, the receiver loses 1 time credit, and the offerer gets one time credit.)

blocked credits: The number of credits that can't be used after applying to an offer.

community: all the members.

event: An event that is offered by a user, that can be applied by other users, free of charge. It is a congregation of site users to socialize.

guest: someone who is not yet a member but is visiting the platform.

lifetime balance: the number of credits that a member has accumulated ssince their registration.

member: someone who has signed up to the platform.

offerer: someone who offers a service.

offer applicant: someone who applied for an offer but their application has not been accepted by the offerer.

offer participant: someone who applied for an offer and their application has been accepted by the offerer.

organizer: the member organizing an event.

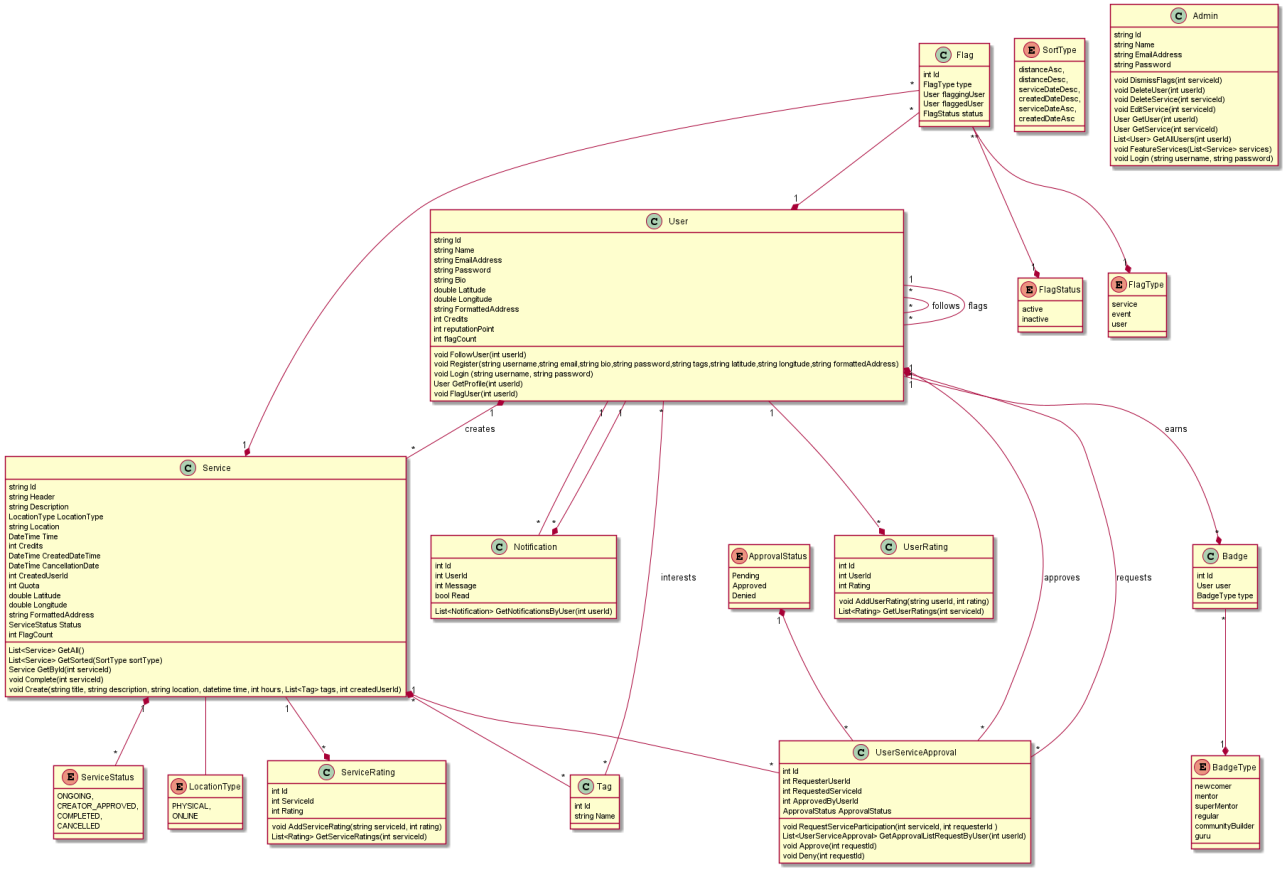
reputation points: points that every member has. They reflect the member's poitive contribution to the community.

service: A meeting that is offered by a user, that can be applied by other users, paid by time currency. The user offers their services, teaching what they know in order to gain time credits.

virtual service: A service that is offered via a zoom link. It does not take place physically.

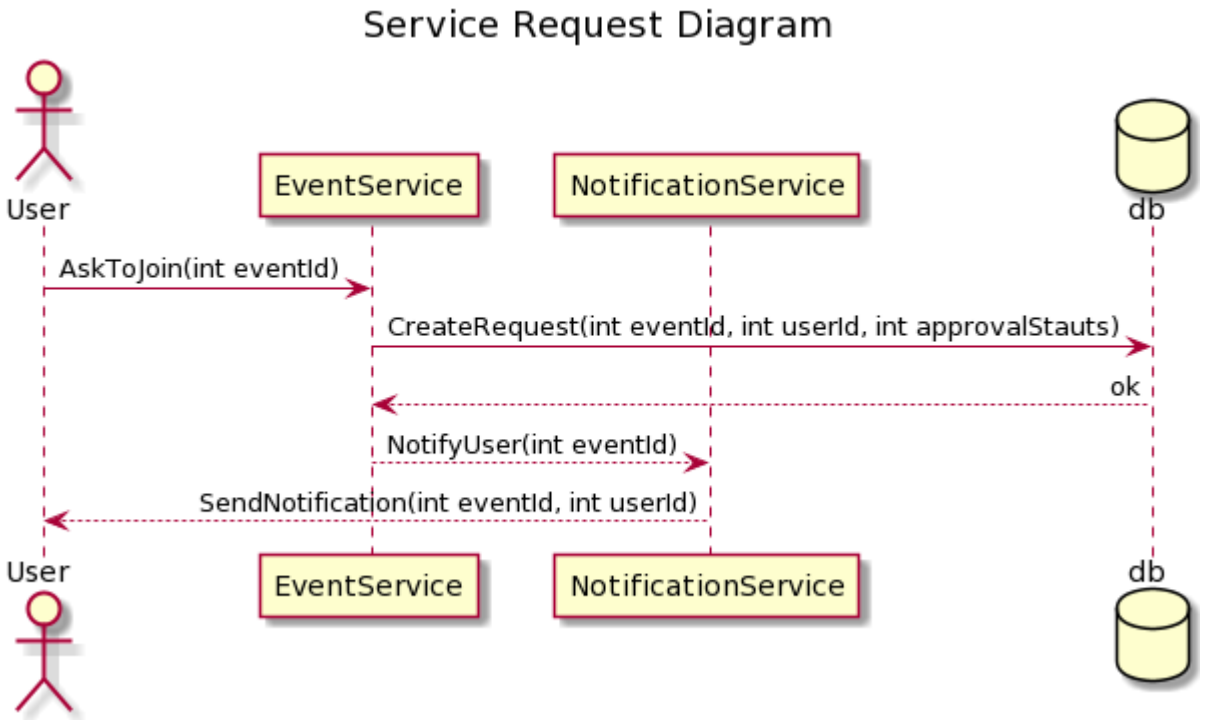
Appendix B: Analysis Models

B.1. Class Diagram

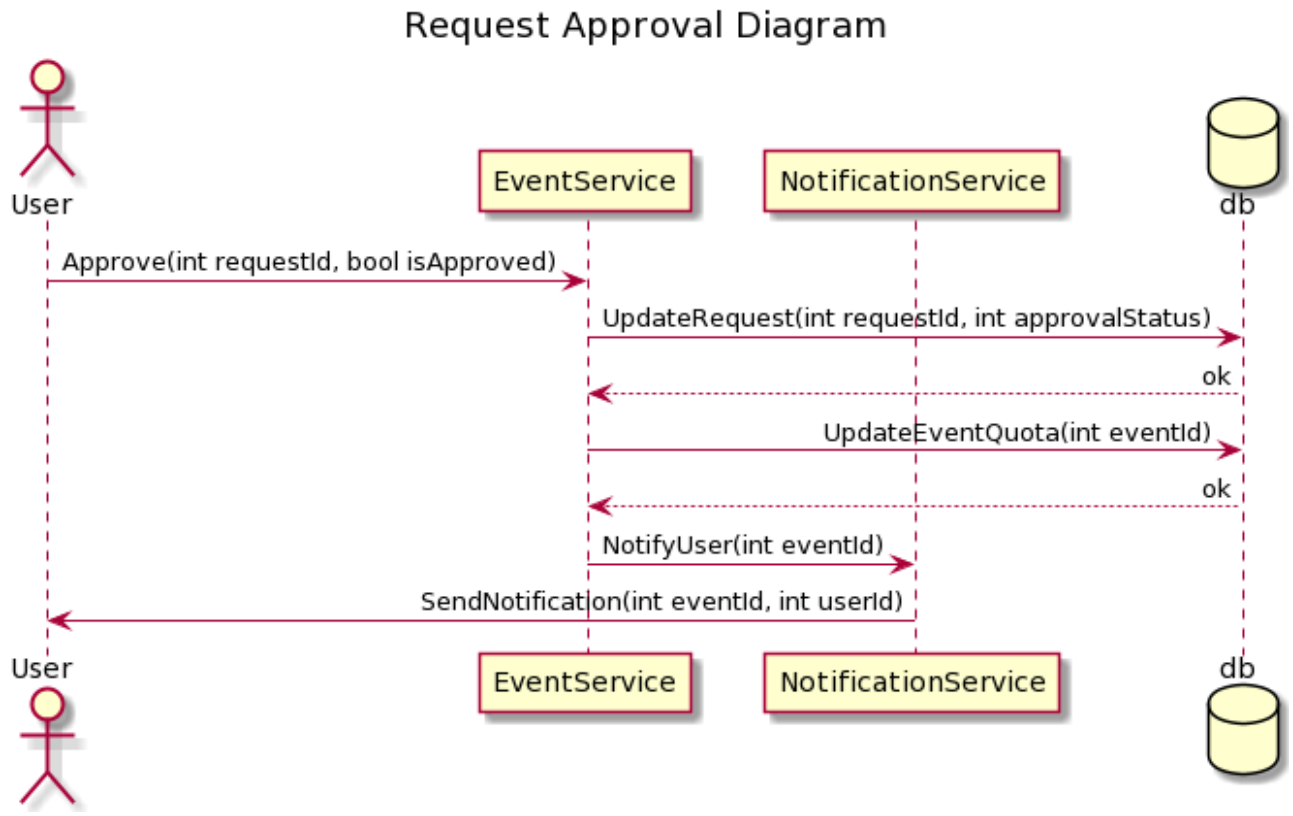


B.2. Sequence Diagrams

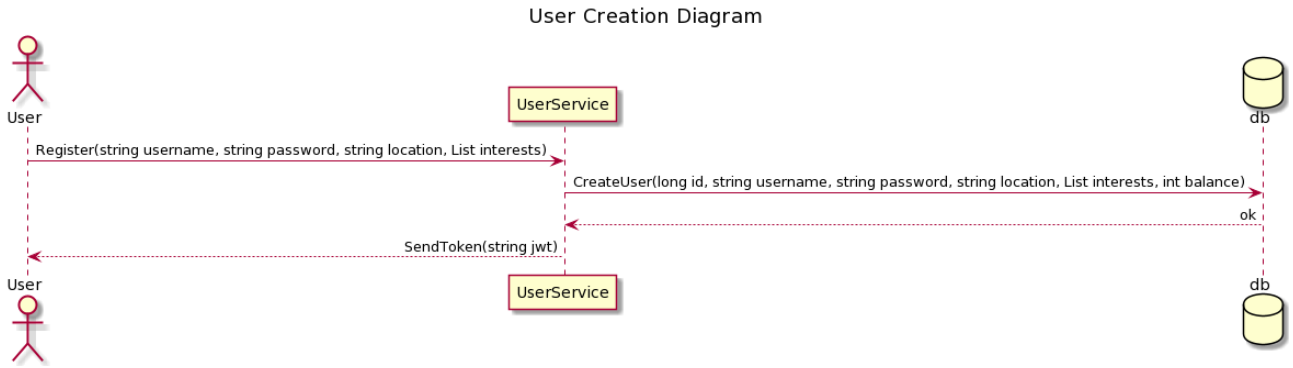
B.2.1. Service Request



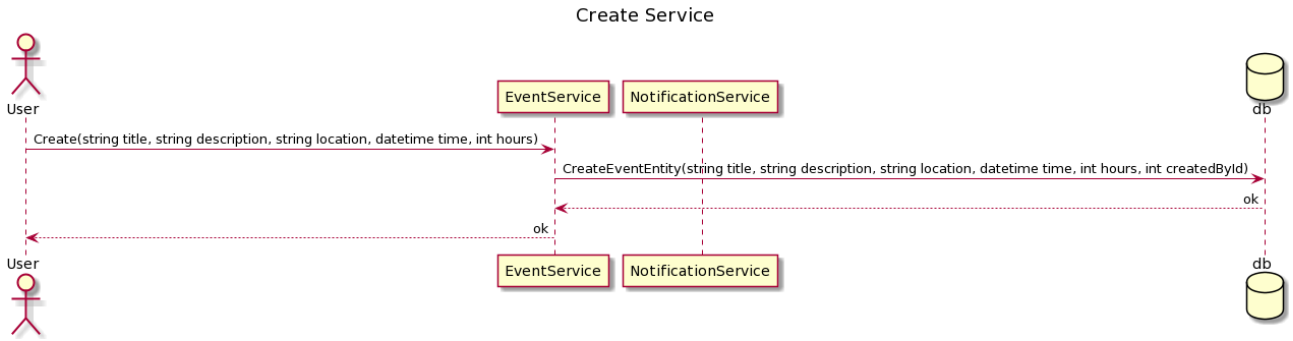
B.2.2. Request Approval



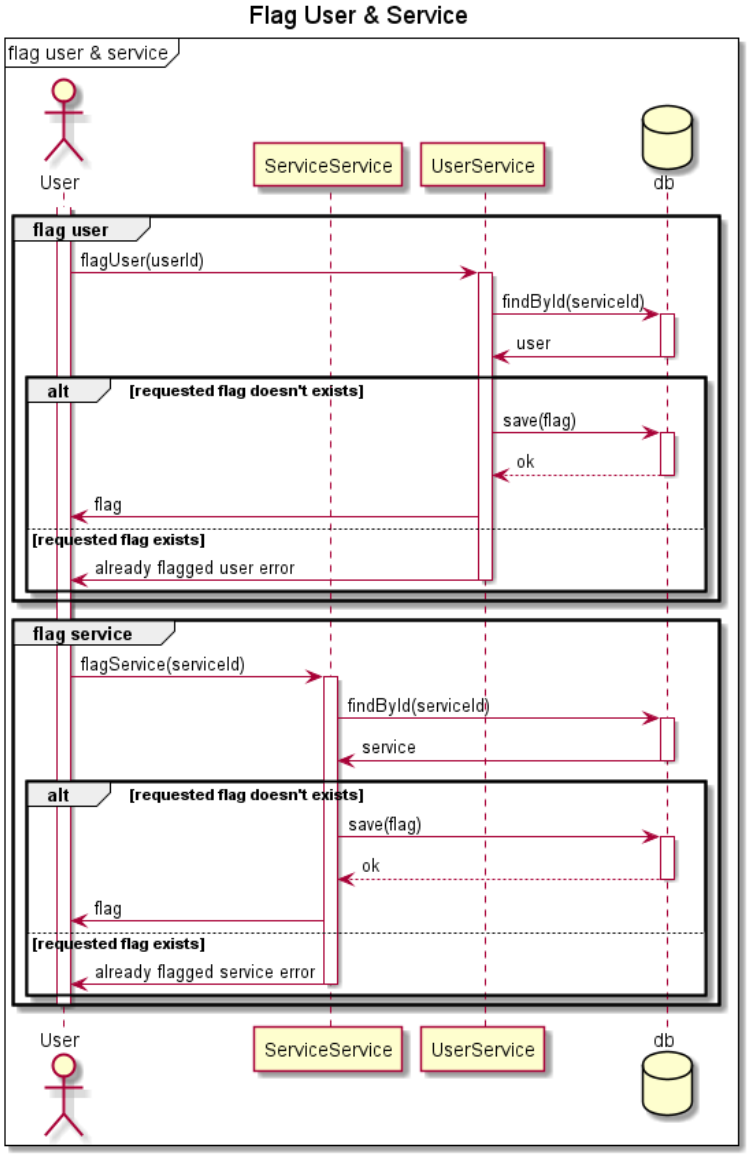
B.2.3. Registration



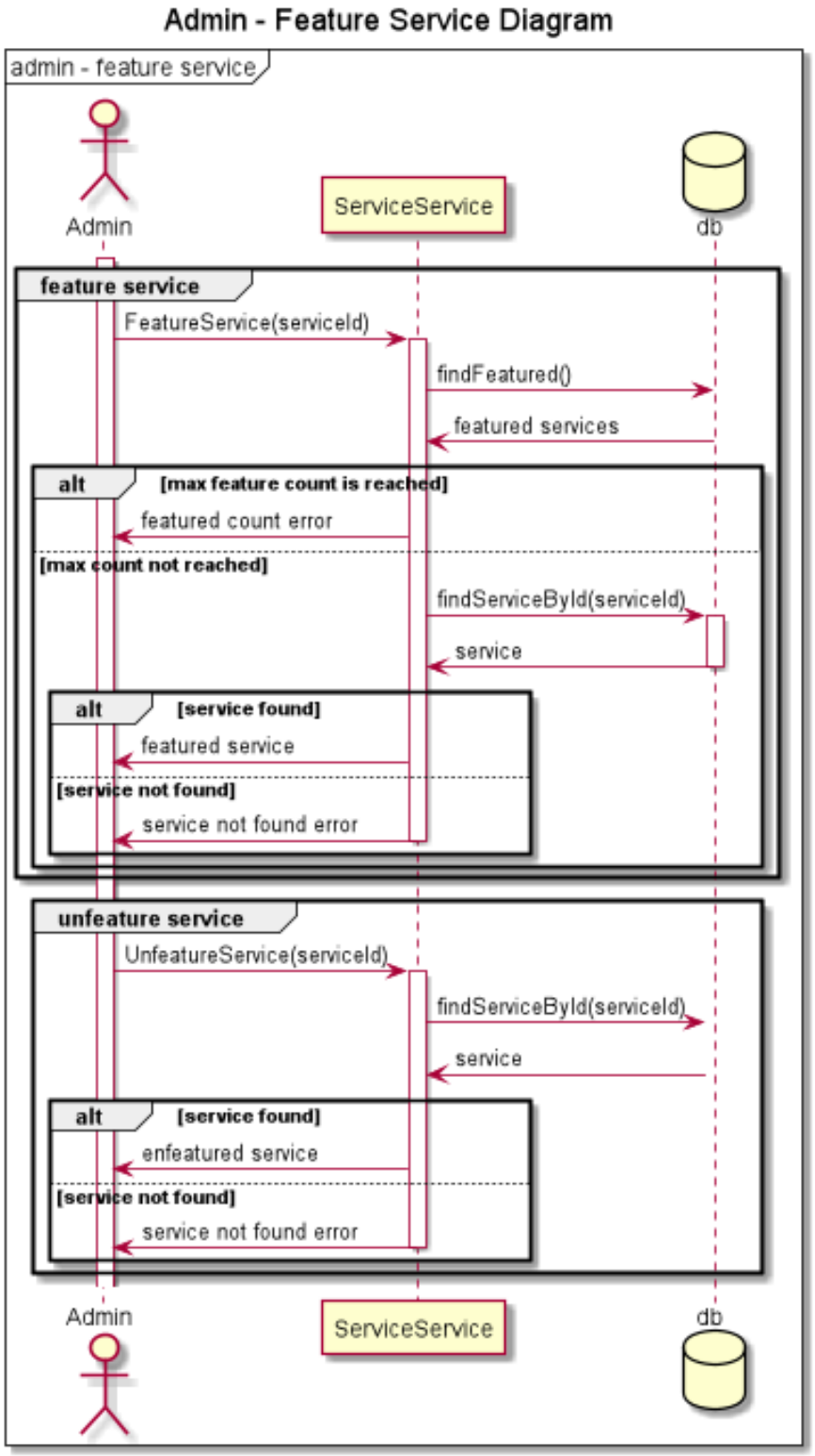
B.2.4. Create Service



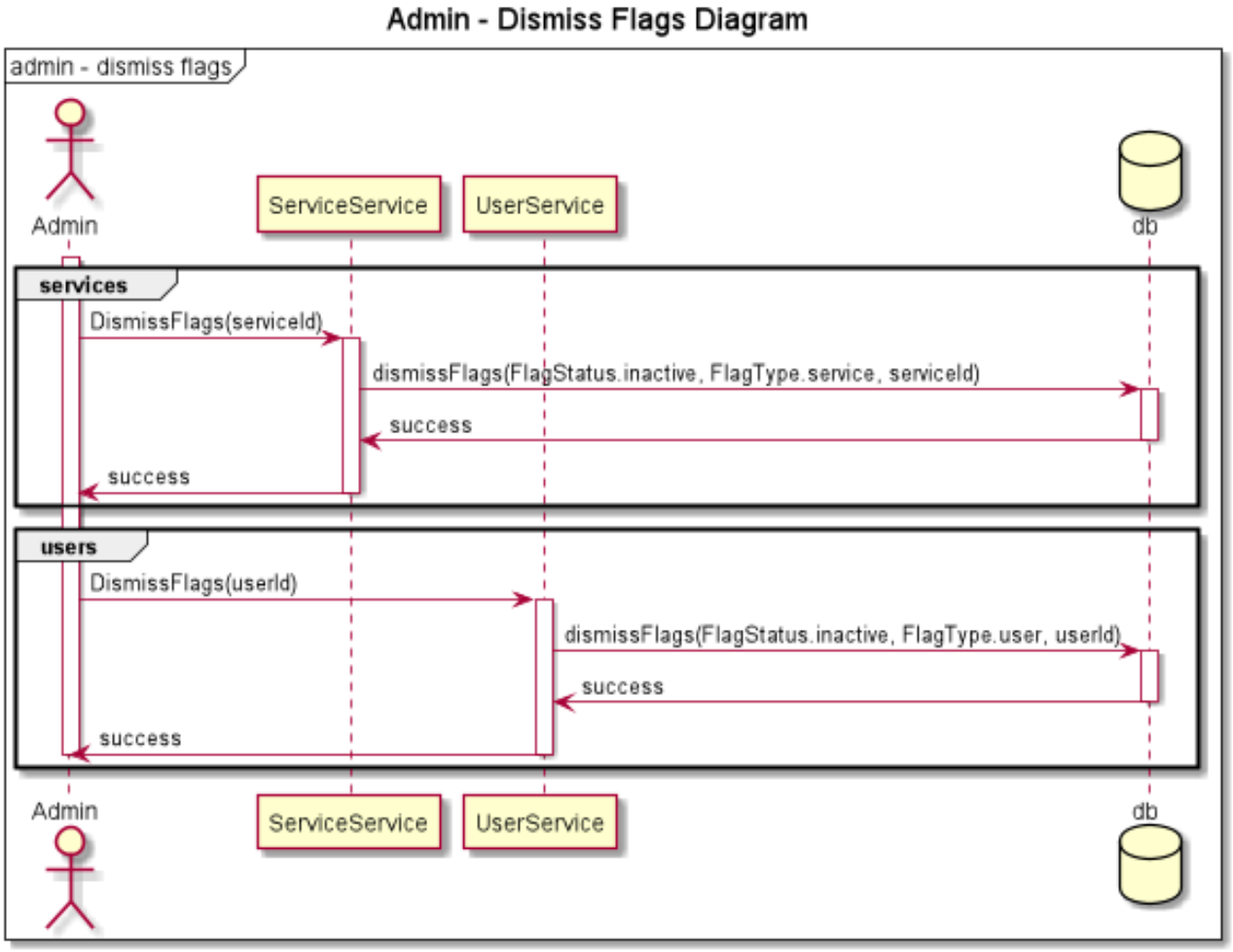
B.2.5. Flag User & Service



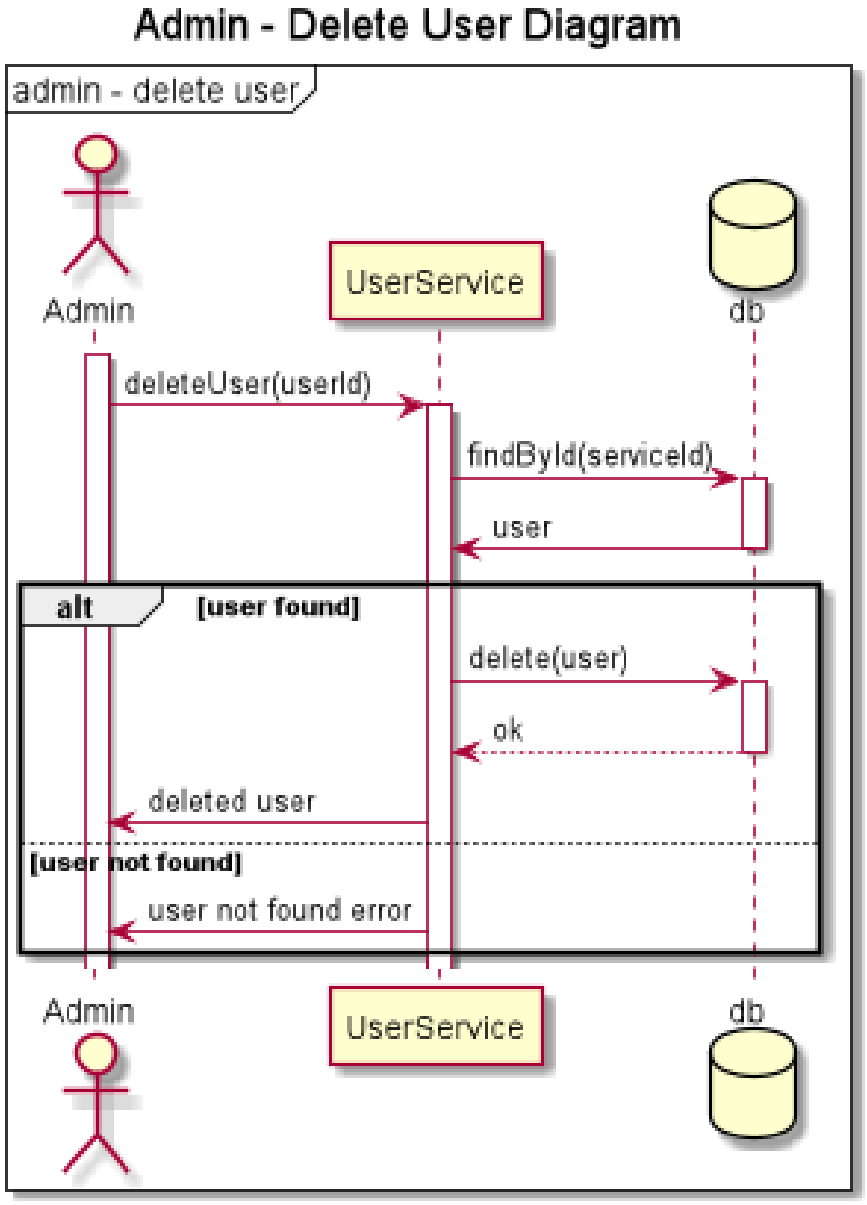
B.2.6. Admin - Feature Service Diagram



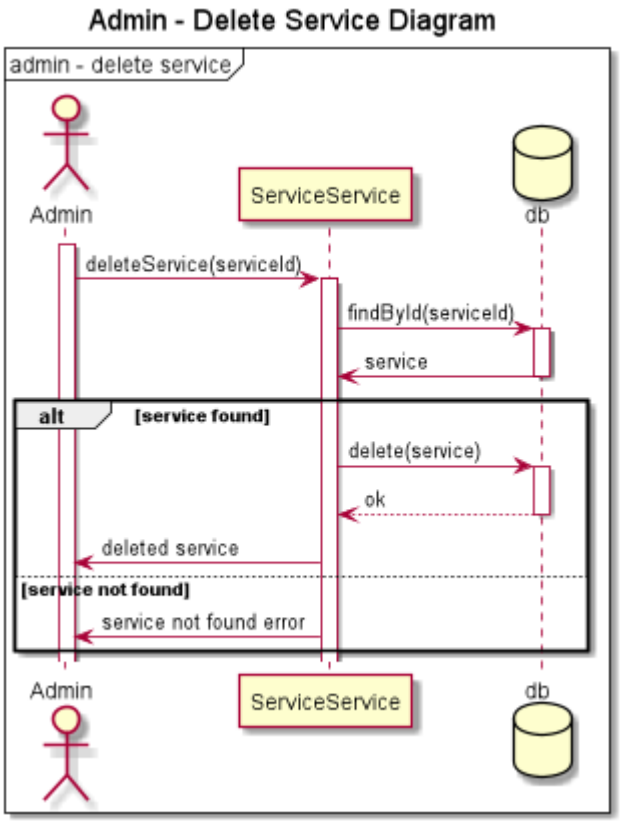
B.2.7. Admin - Dismiss Flag Diagram



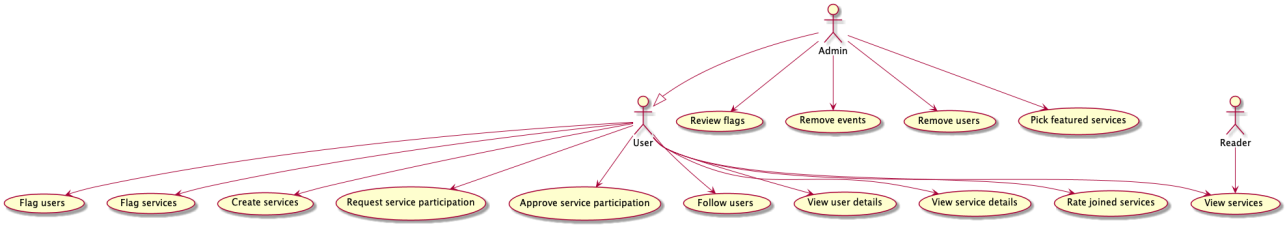
B.2.8. Admin - Delete User Diagram



B.2.9. Admin - Delete Service Diagram

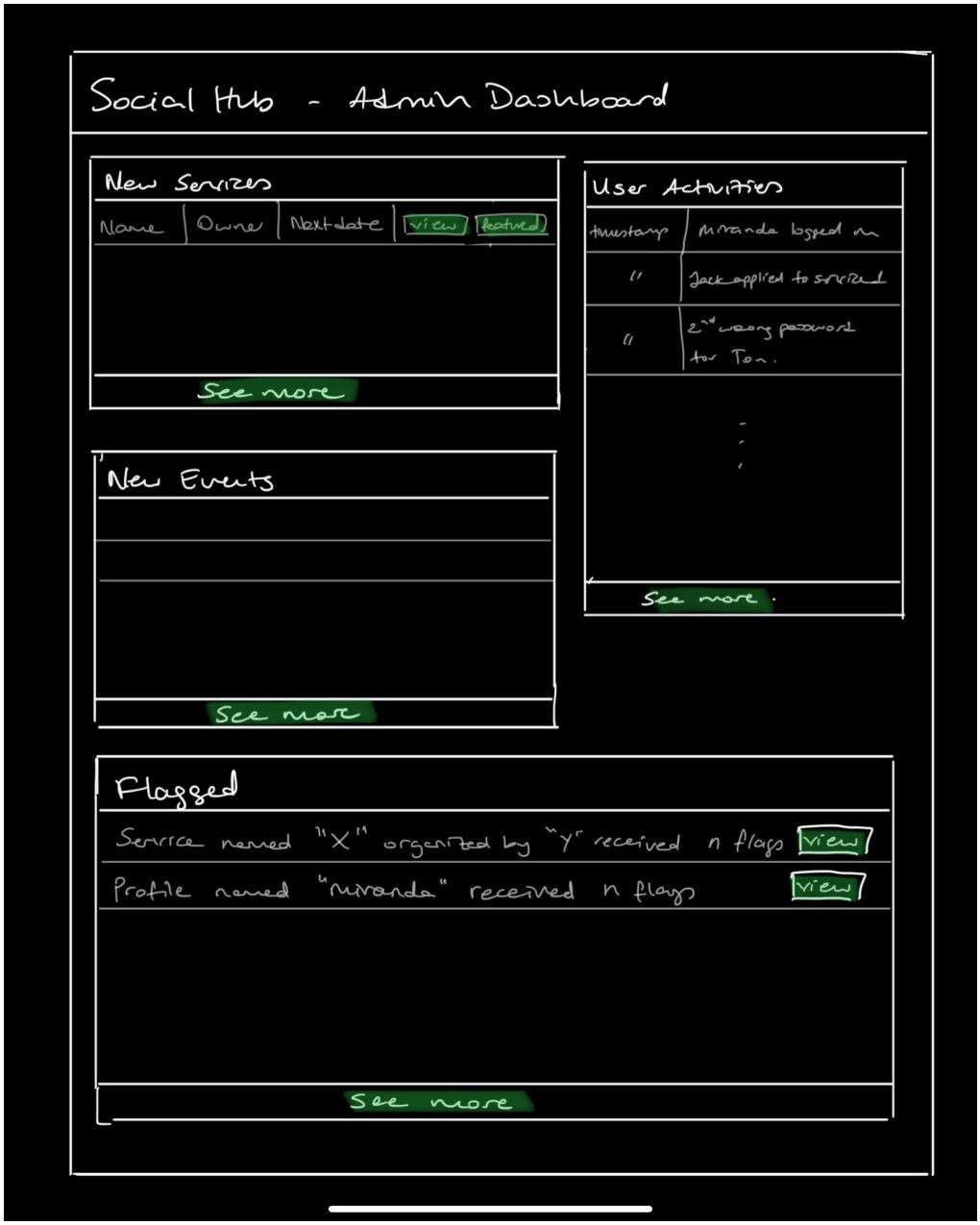


B.3. Use Case Diagrams

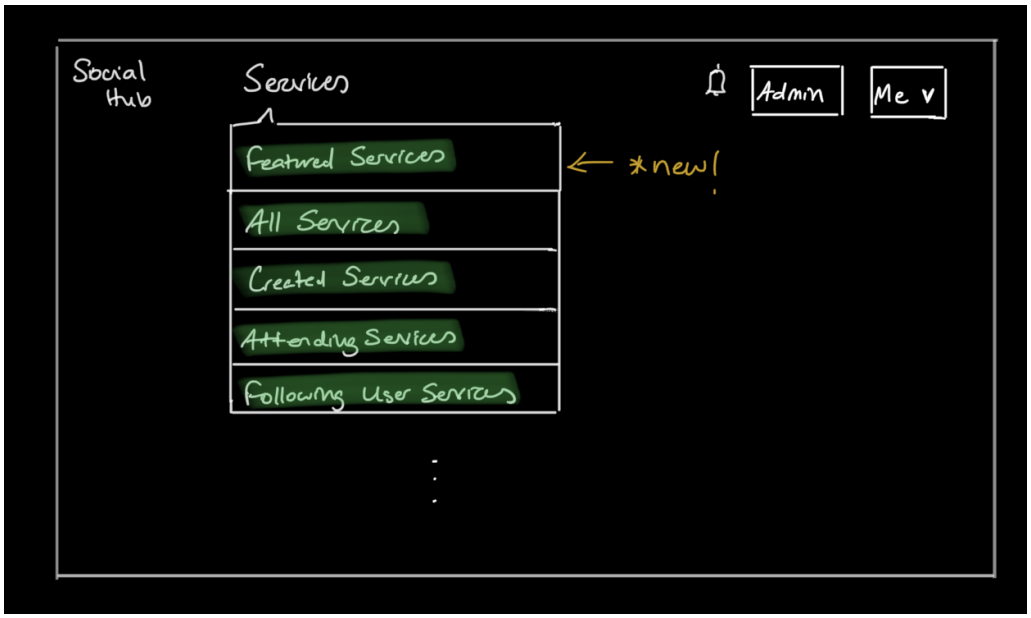


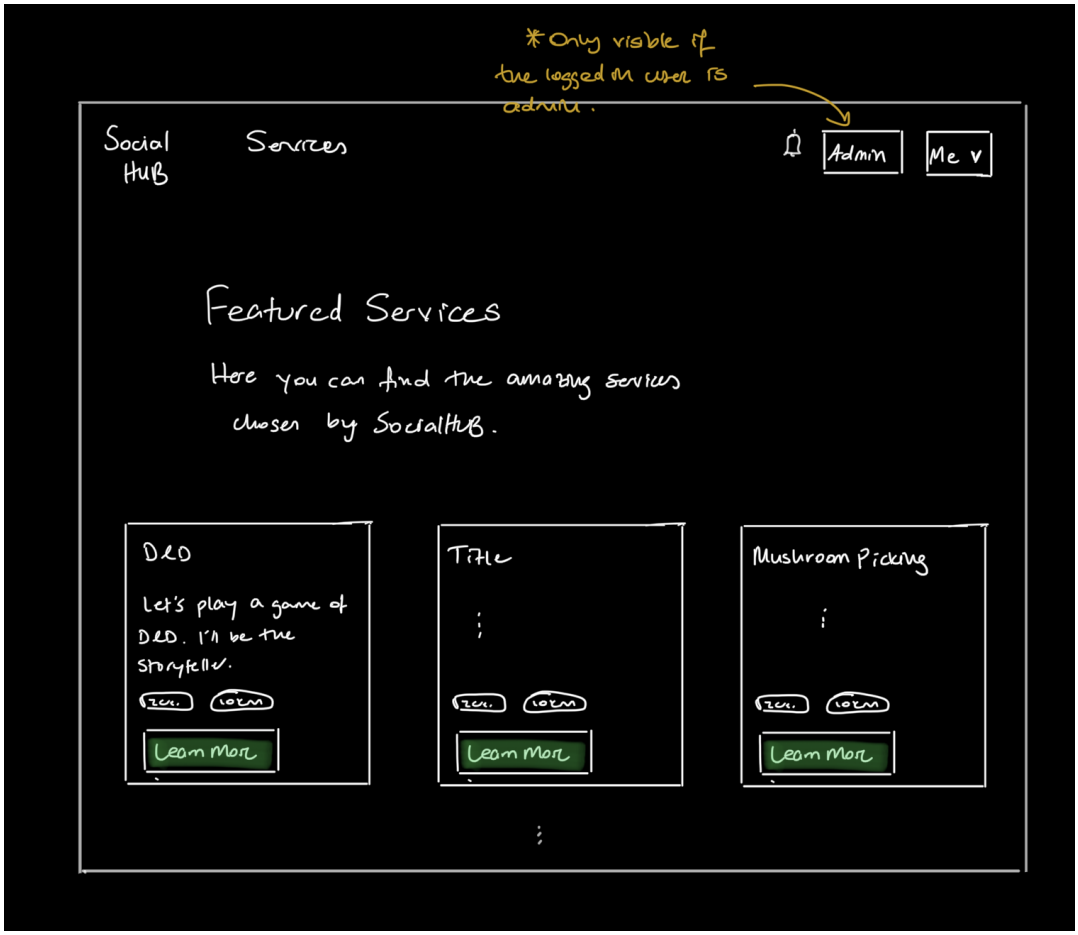
B.4. Mockups

B.4.1. Admin Dashboard



B.4.2. Featured Services Mockup





Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>