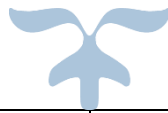




INFORMATICS 3A10

Final Year Project



Group Name	The Overclockers
Group Number	25
Project Name	Eternal Vibes
Name of Project Supervisor	Brian Greaves

Name	Surname	Student Number
Ronald	Lai	201433999
David	Innes	201474149
Devandrin	Kuni	201320596
Marc	Zuze	201477488

Plagiarism Document



Academy of Computer Science and Software Engineering

Informatics 3 Group Project

Plagiarism Declaration

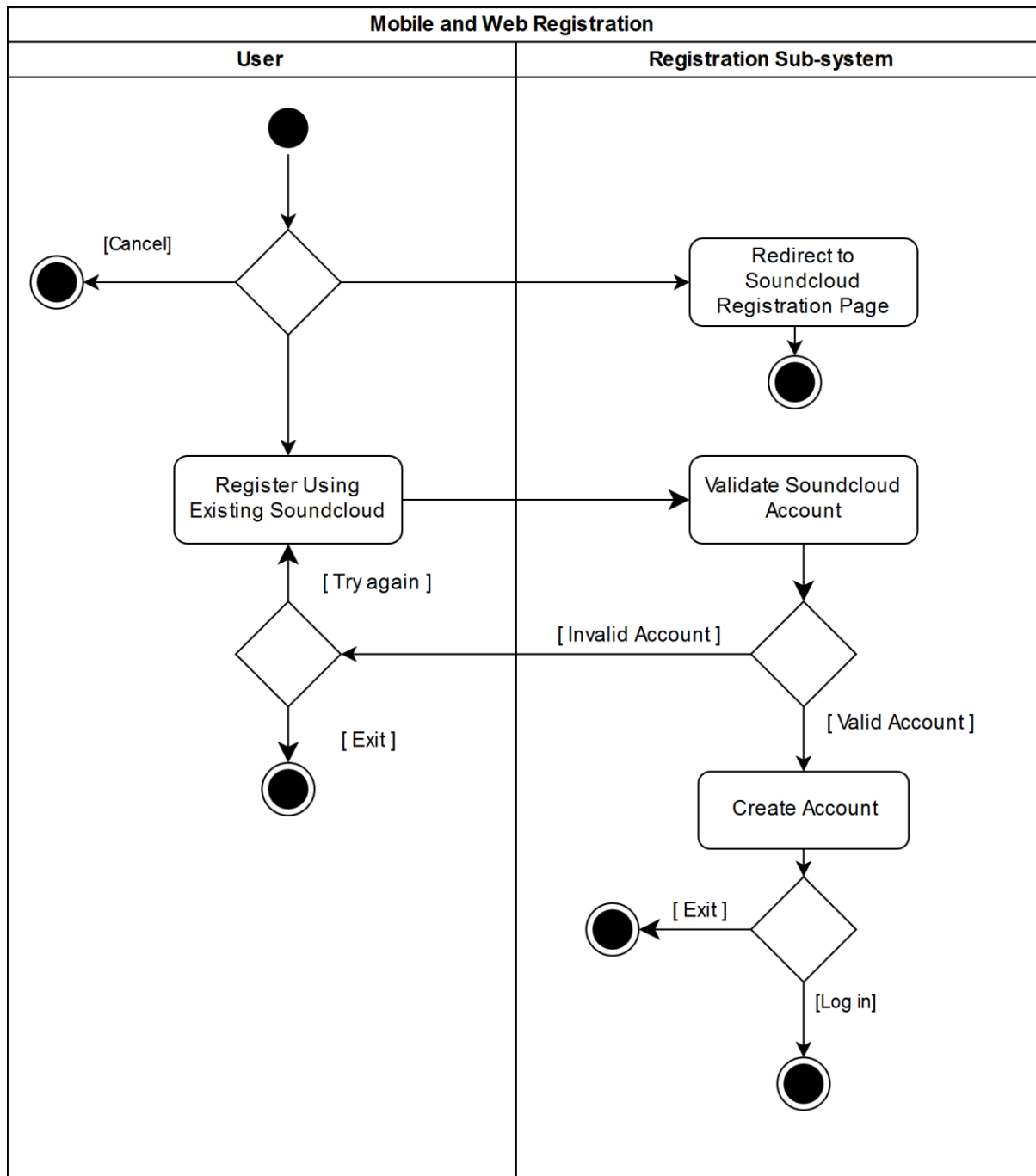
Deliverable #	2	Team #	25	Supervisor	Brian Greaves
Team Name	The Overclockers				

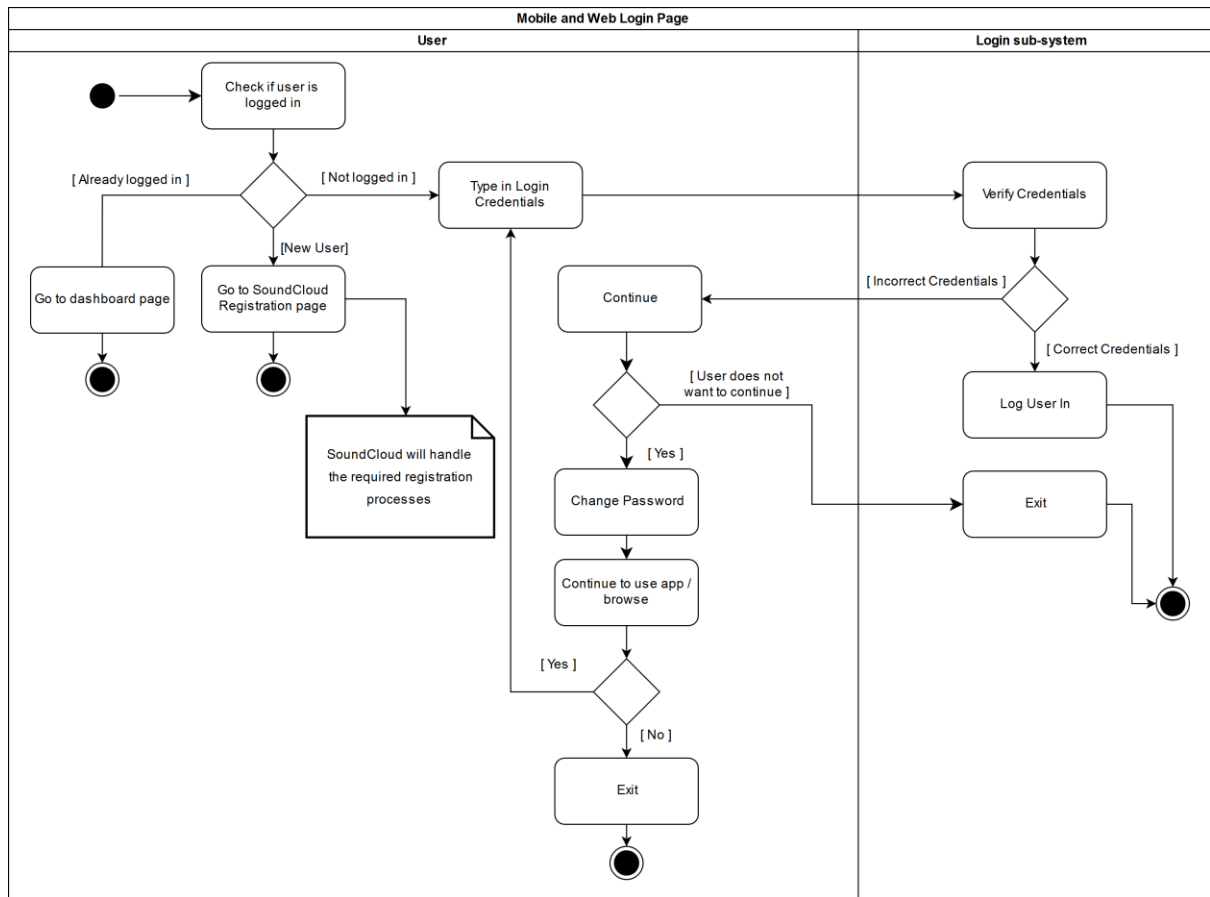
1. Plagiarism is to present someone else's ideas as our own.
2. Where material created by other people has been used (either from a printed or digital source) this has been carefully acknowledged and referenced. We have used the appropriate style for citation and referencing. Every contribution to and quotation from the work of other people in this deliverable has been acknowledged through citation and reference.
3. We know that plagiarism is wrong.
 - 3.1. We understand what plagiarism is and am aware of the University's policy in this regard.
 - 3.2. We know that we would plagiarise if we do not give credit to my sources, or if we copy any part of a book, article, or Internet source without proper citation.
 - 3.3. We know that even if we only change the copied work slightly, we still plagiarise when using someone else's work without proper citation.
 - 3.4. We declare that we have created our own work throughout this deliverable and we have credited all ideas we have gained from other people's work.
4. We declare that this deliverable is our own original work.
5. We have not allowed, and will not allow, anyone to copy our work with the intention of passing it off as his or her own work.

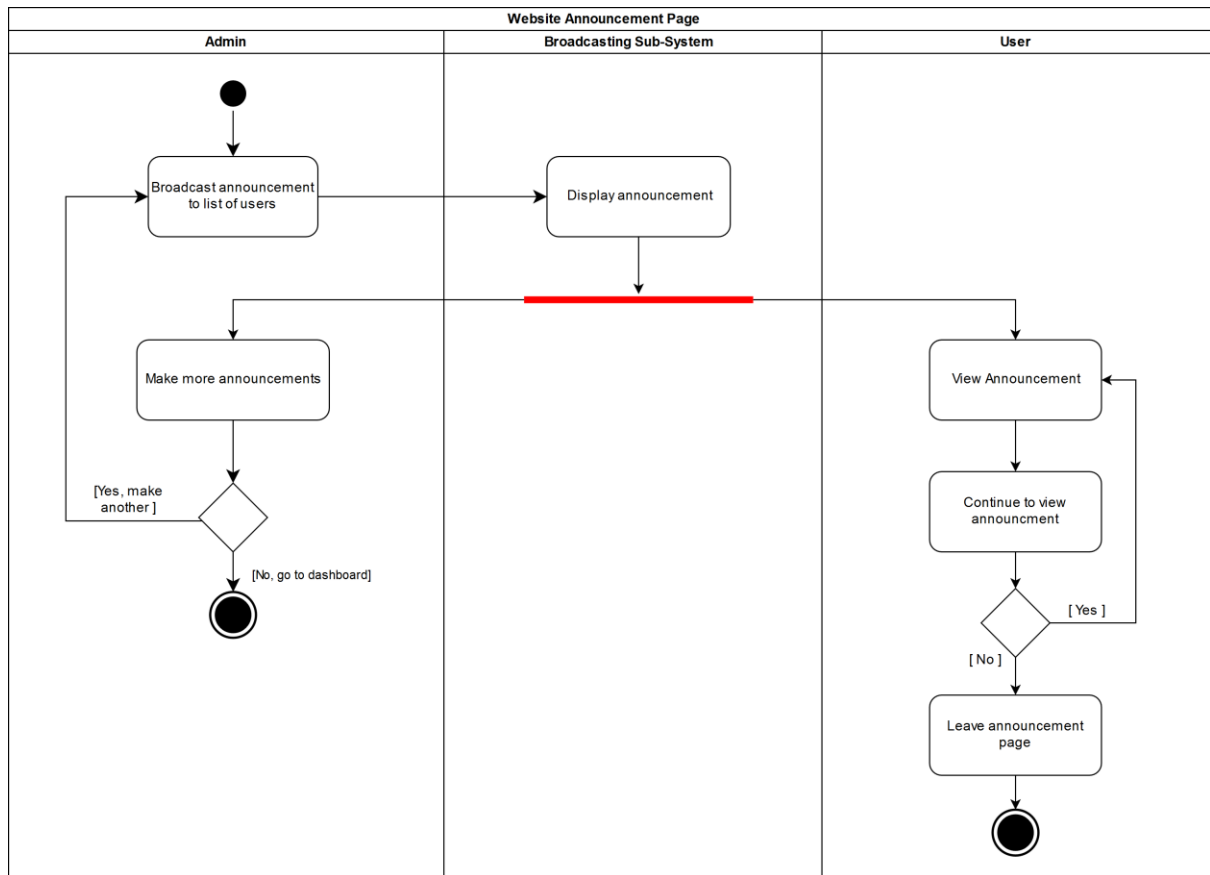
Signature		Signature	
Full Name	Marco Lermanzo Zuze	Full Name	Devandrin Kuni
Student #	201477488	Student #	201320596
Date	29/03/2017	Date	29/03/2017
Member 1		Member 2	

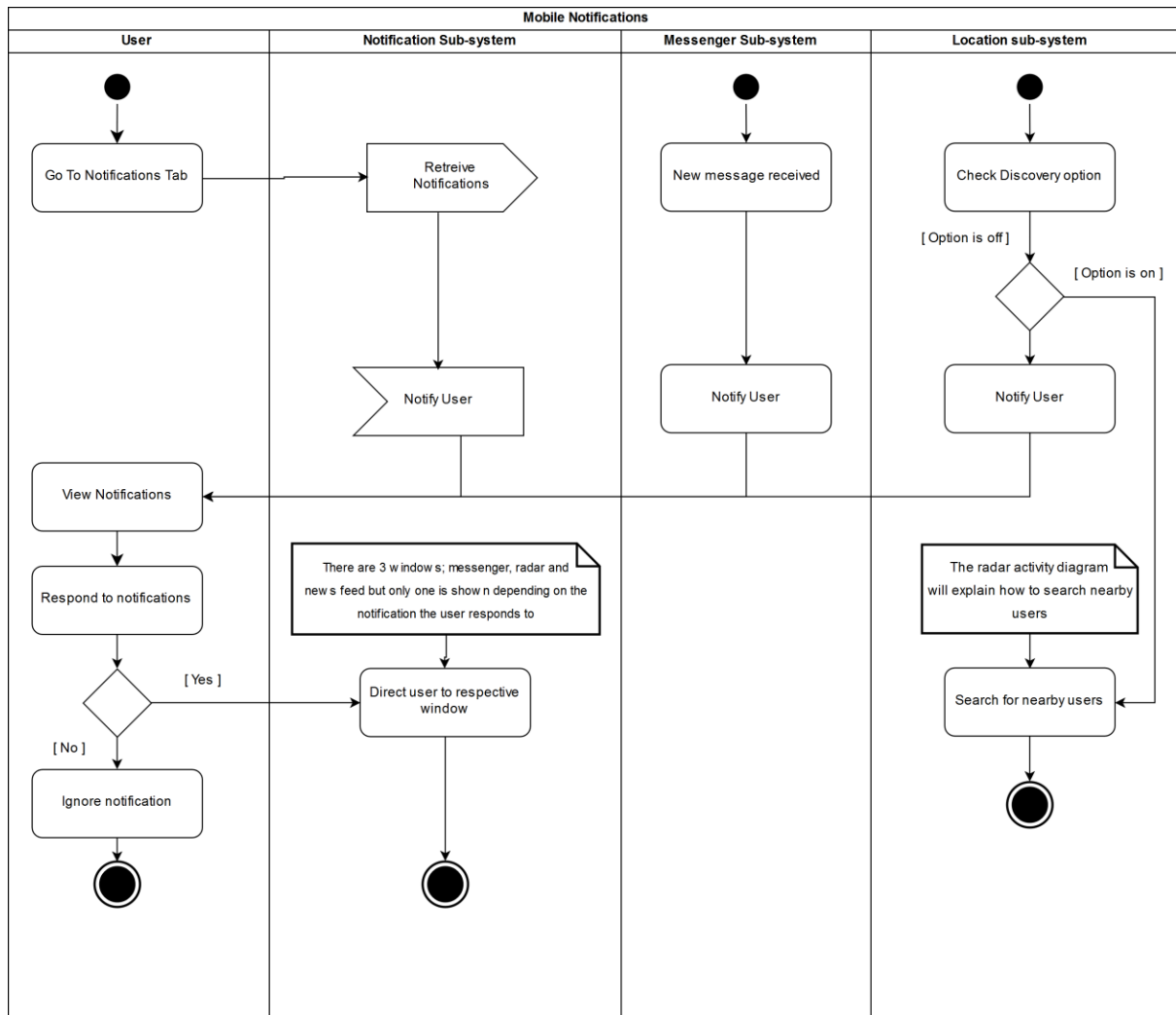
Signature		Signature	
Full Name	David Innes	Full Name	Ronald Lai
Student #	201474149	Student #	201433999
Date	29/03/2017	Date	29/03/2017
Member 3		Member 4	

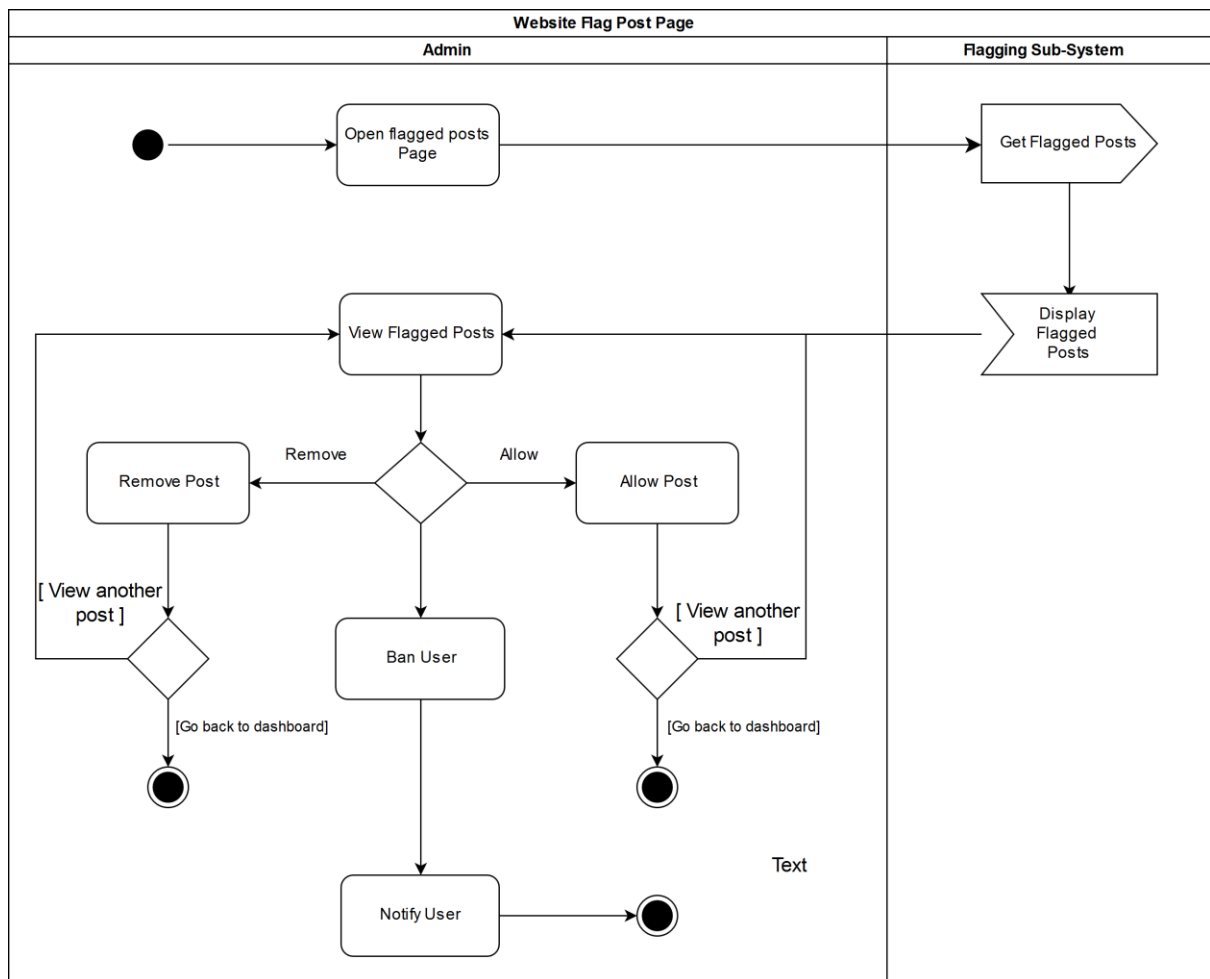
Activity Diagrams

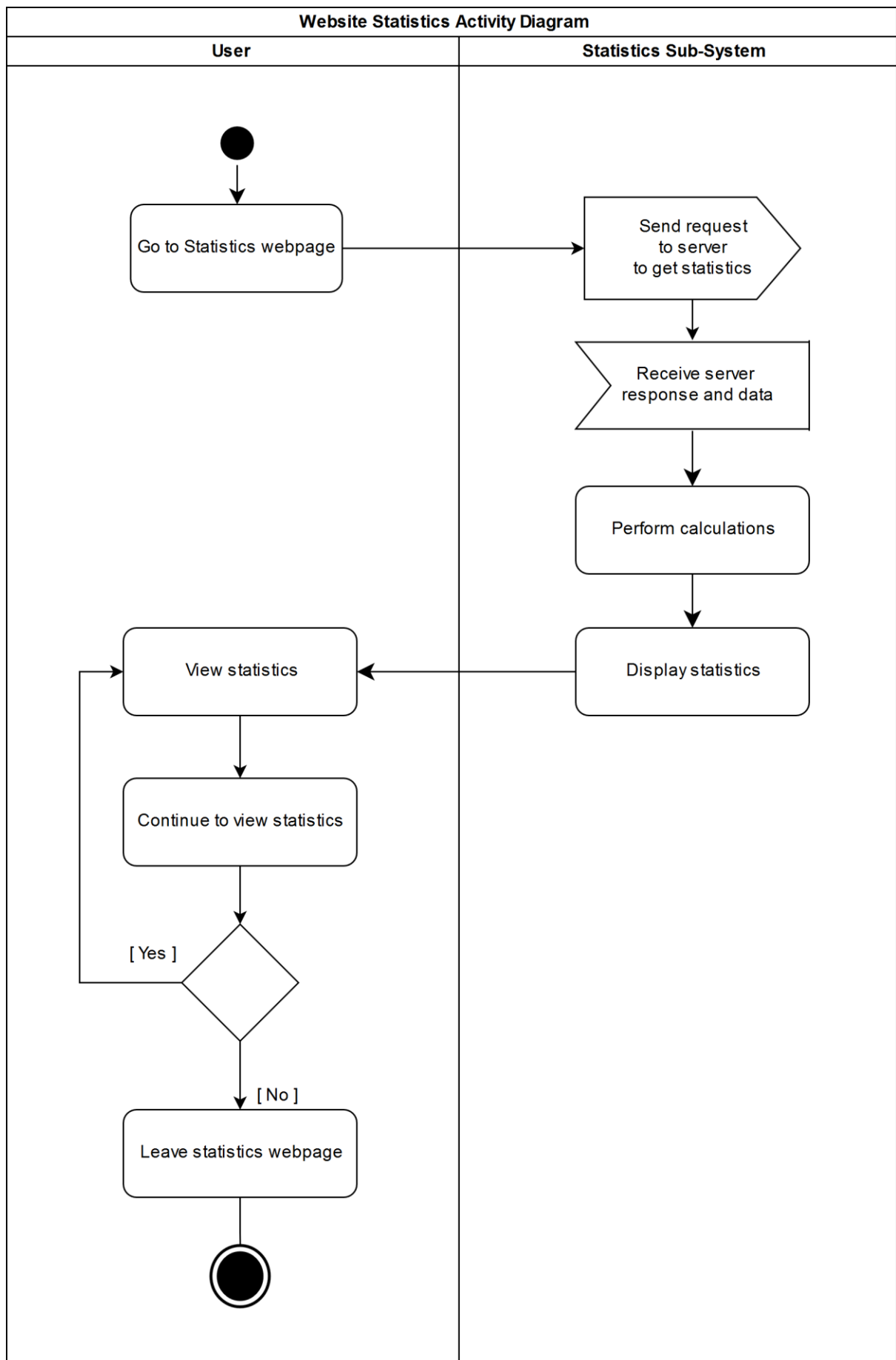


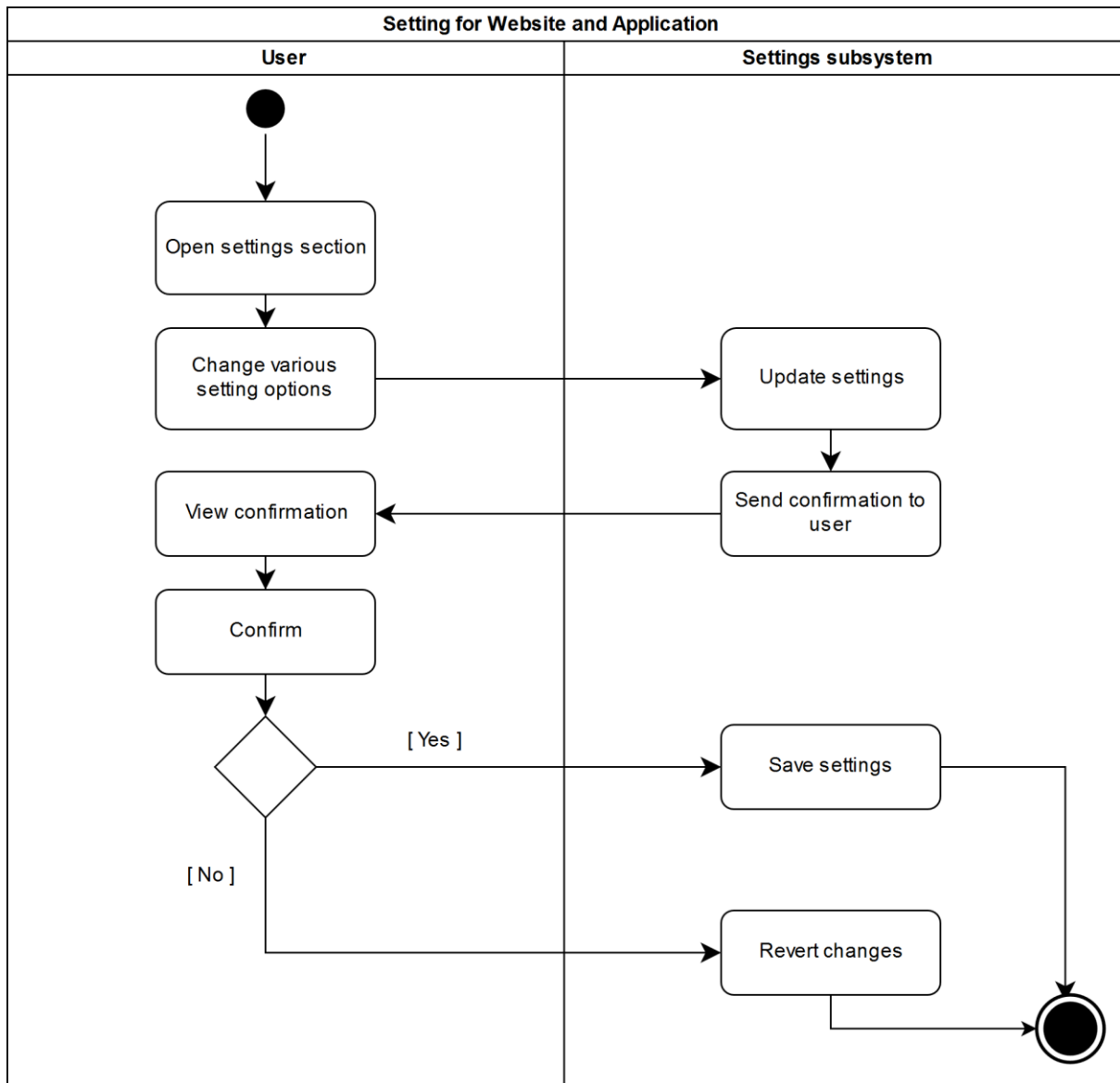


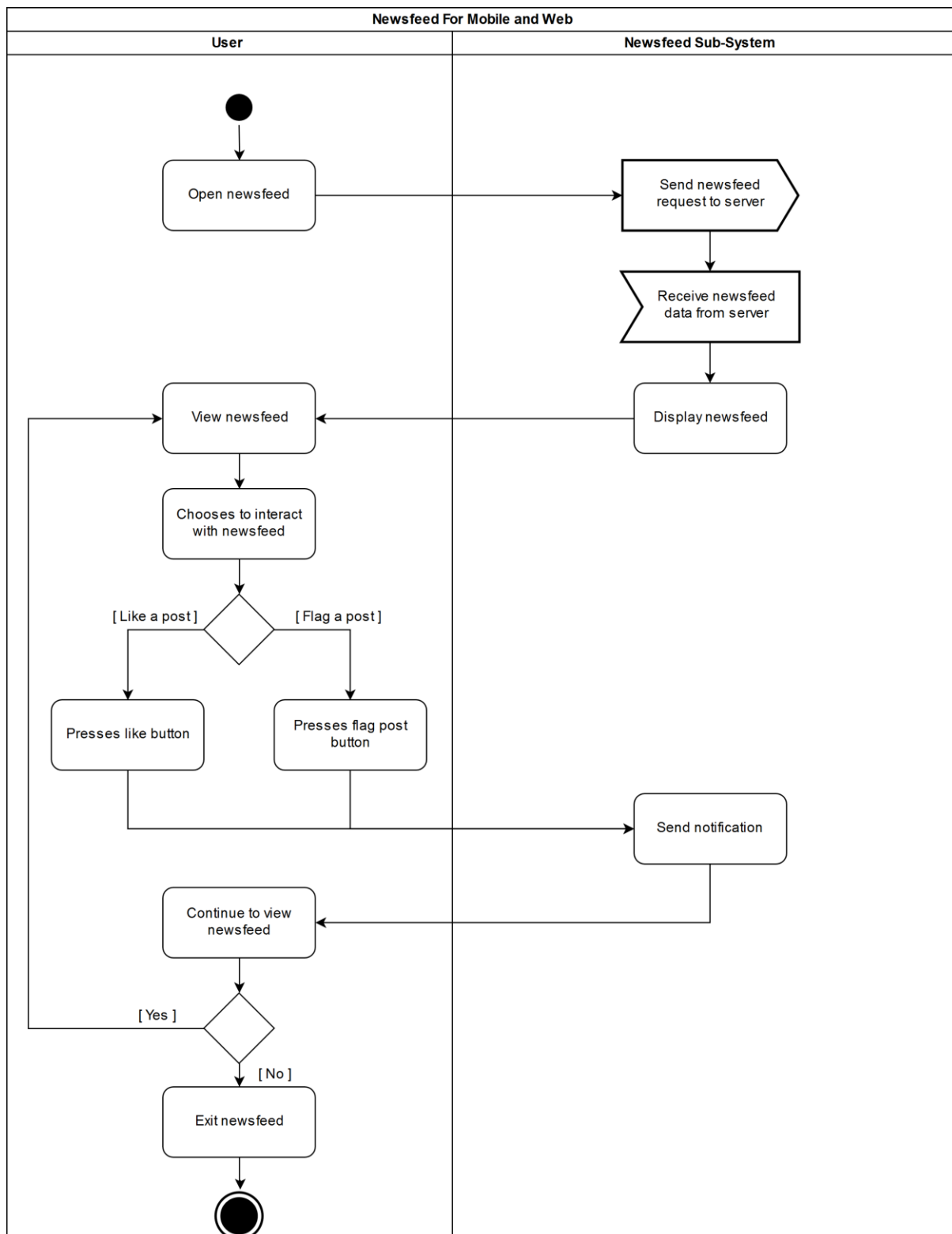


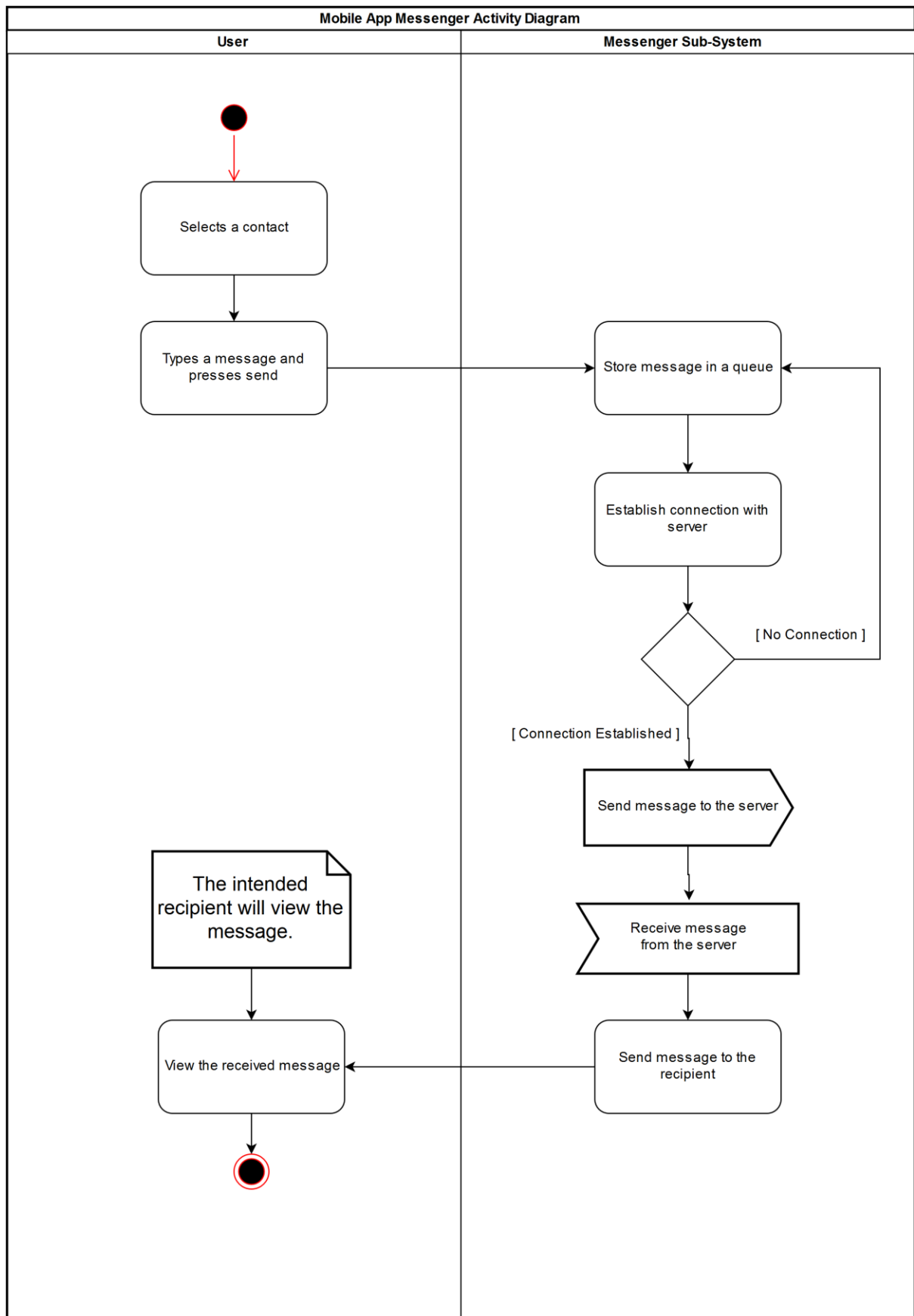


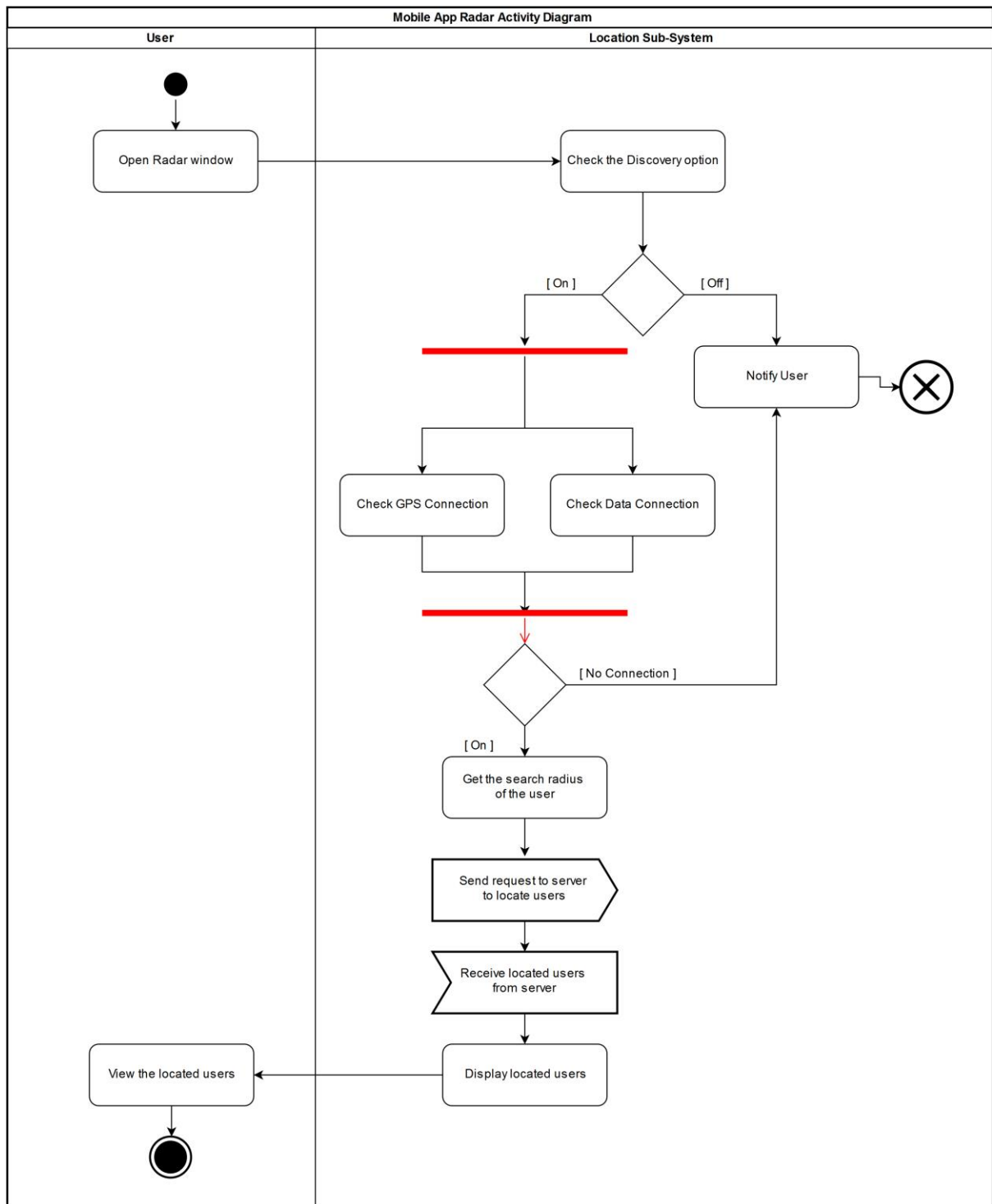




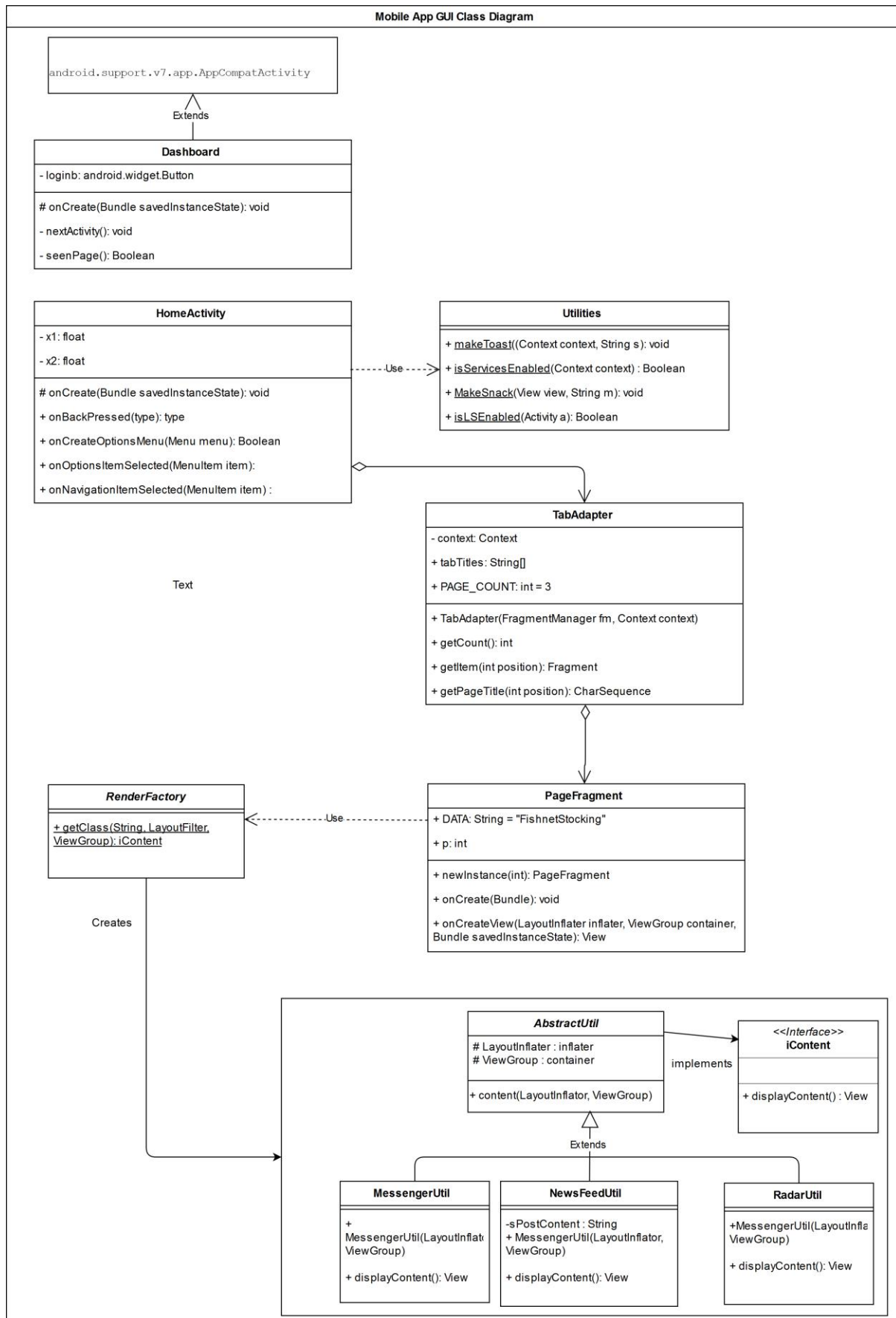


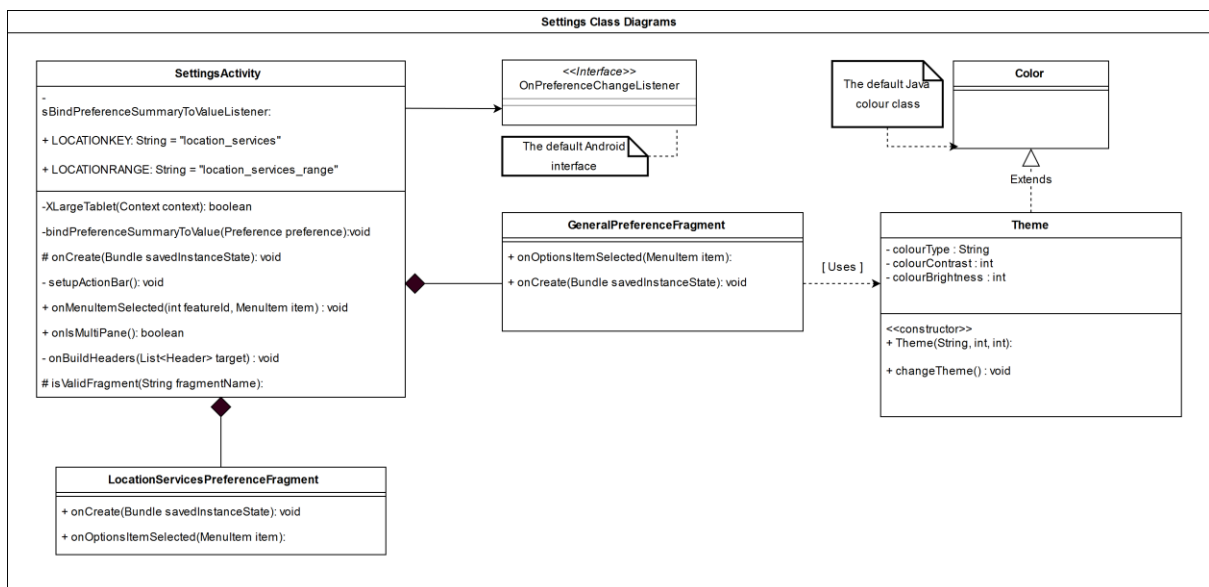
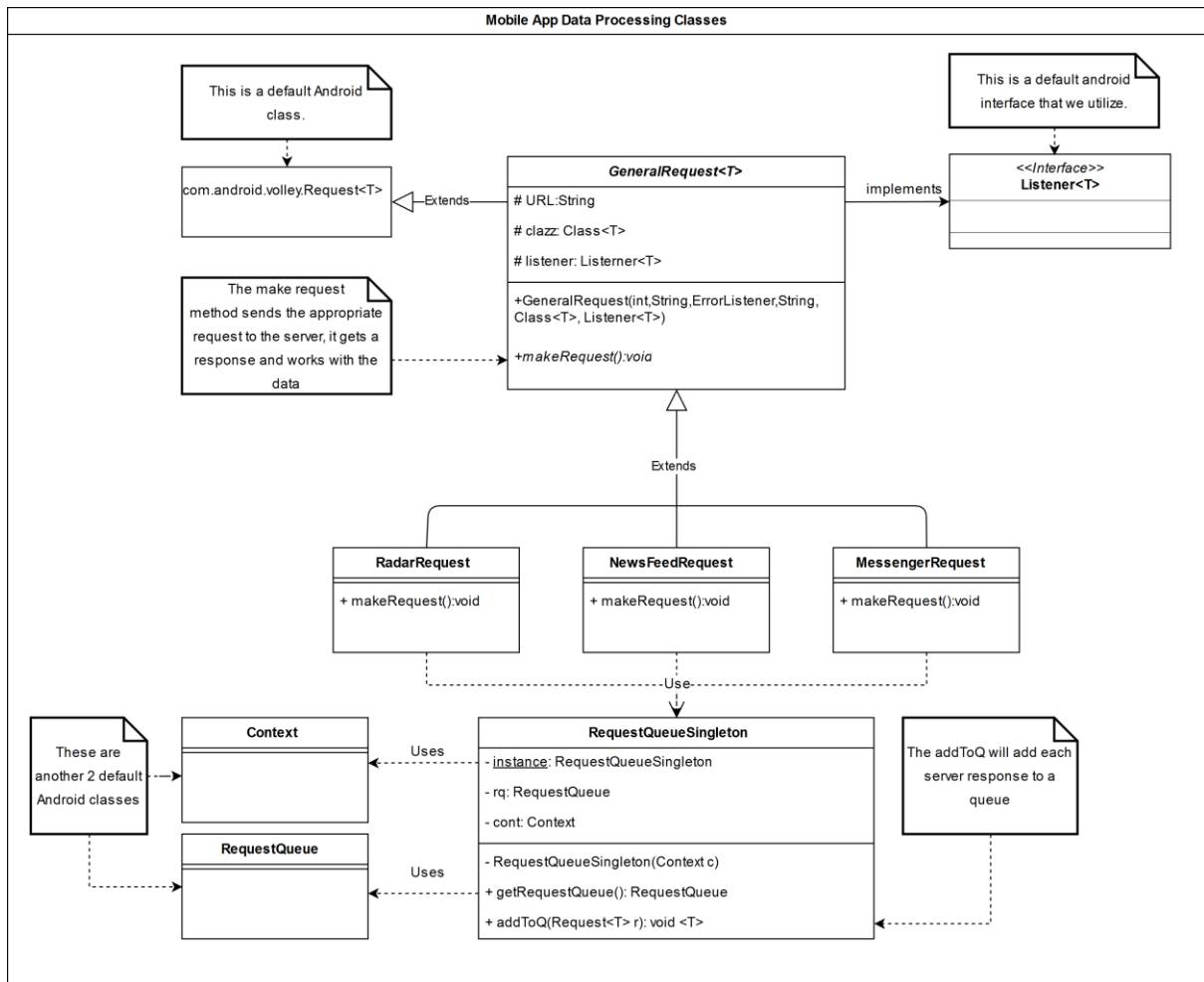


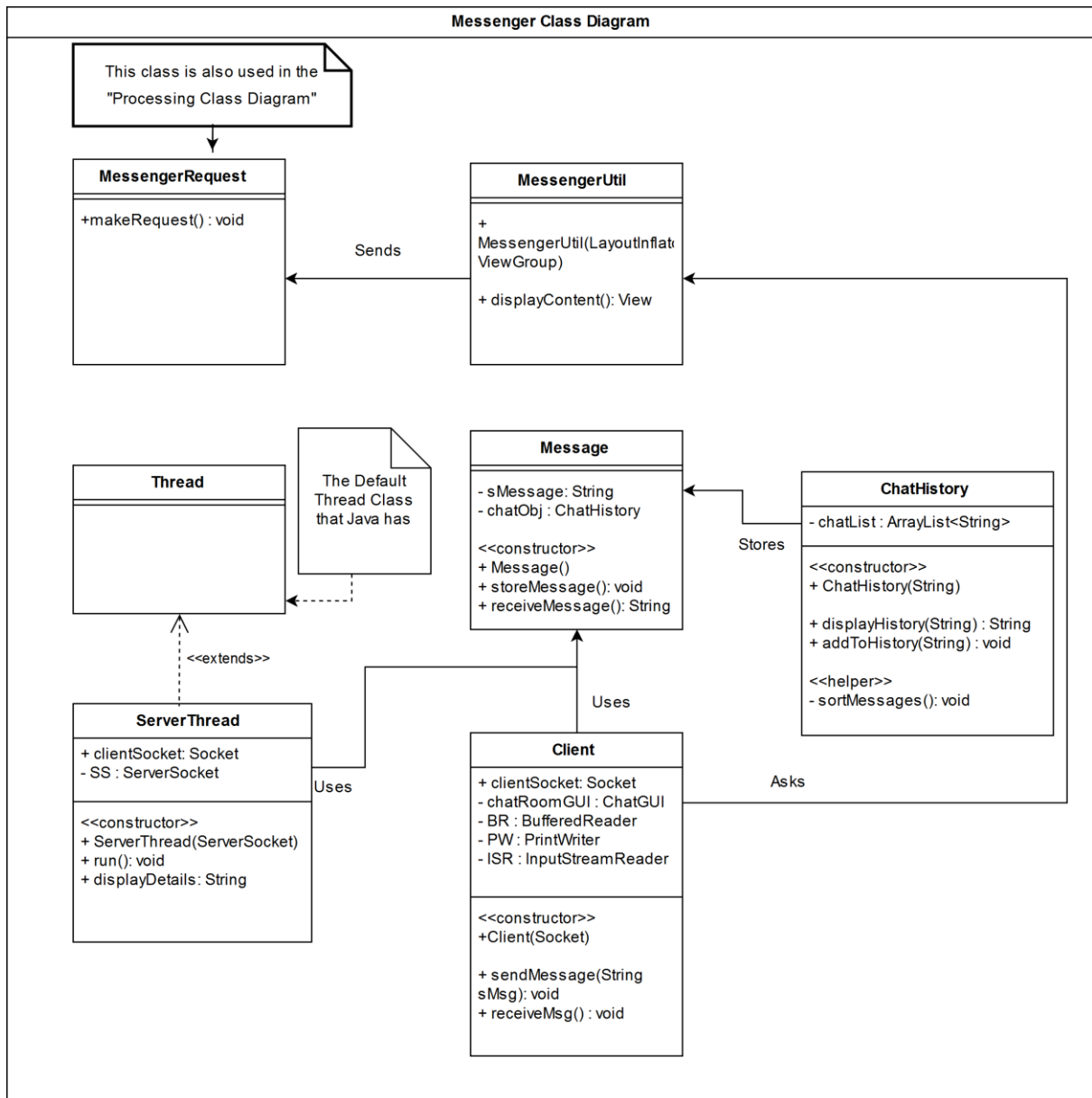


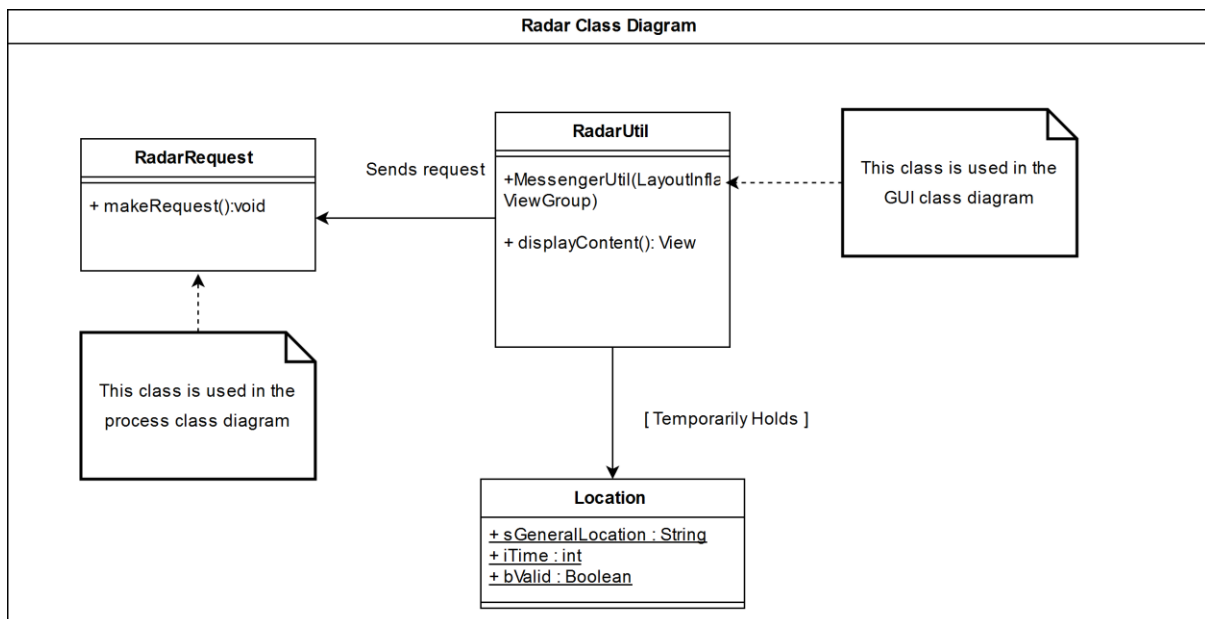
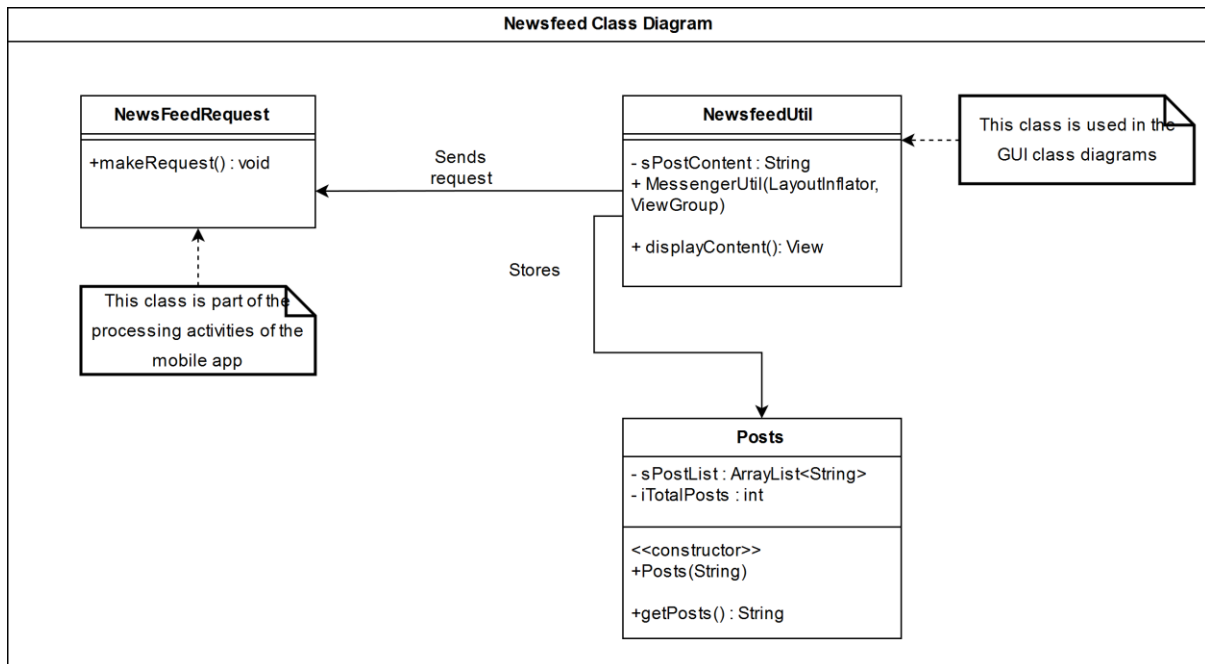


Class Diagrams



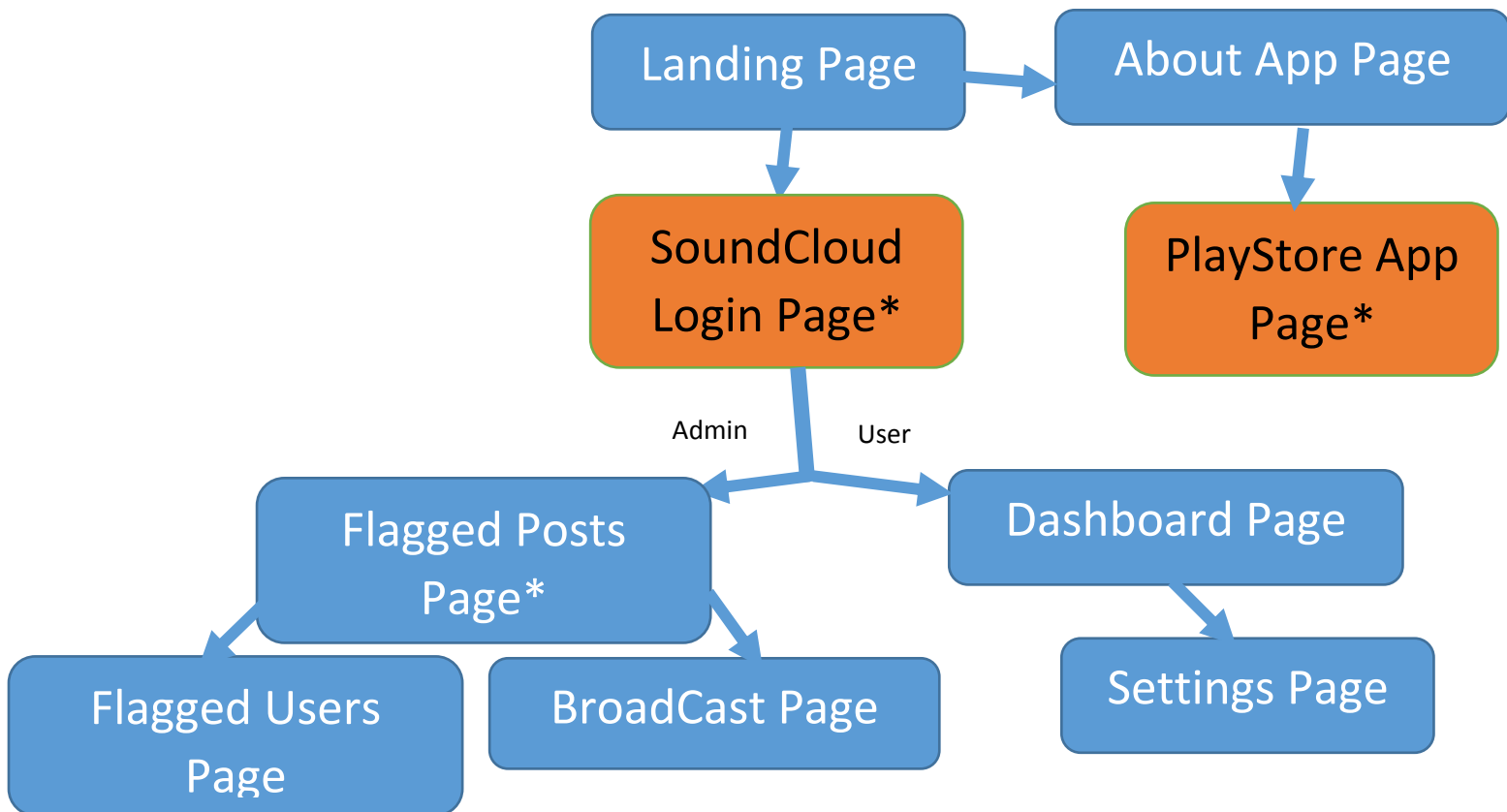






Prototype Interfaces

Website Map



* The SoundCloud page will either redirect the user to the Admin section or the User section (Dashboard Page) depending on if the ID that SoundCloud gives us is for a User or an Admin (A decision made by a query to our database with SoundCloud's response), the default page for the Admin Section is the Flagged Posts Page, due to it being the page with the most activity from the admin, the other pages are then of weighting and can be accessed by the navigation bar.

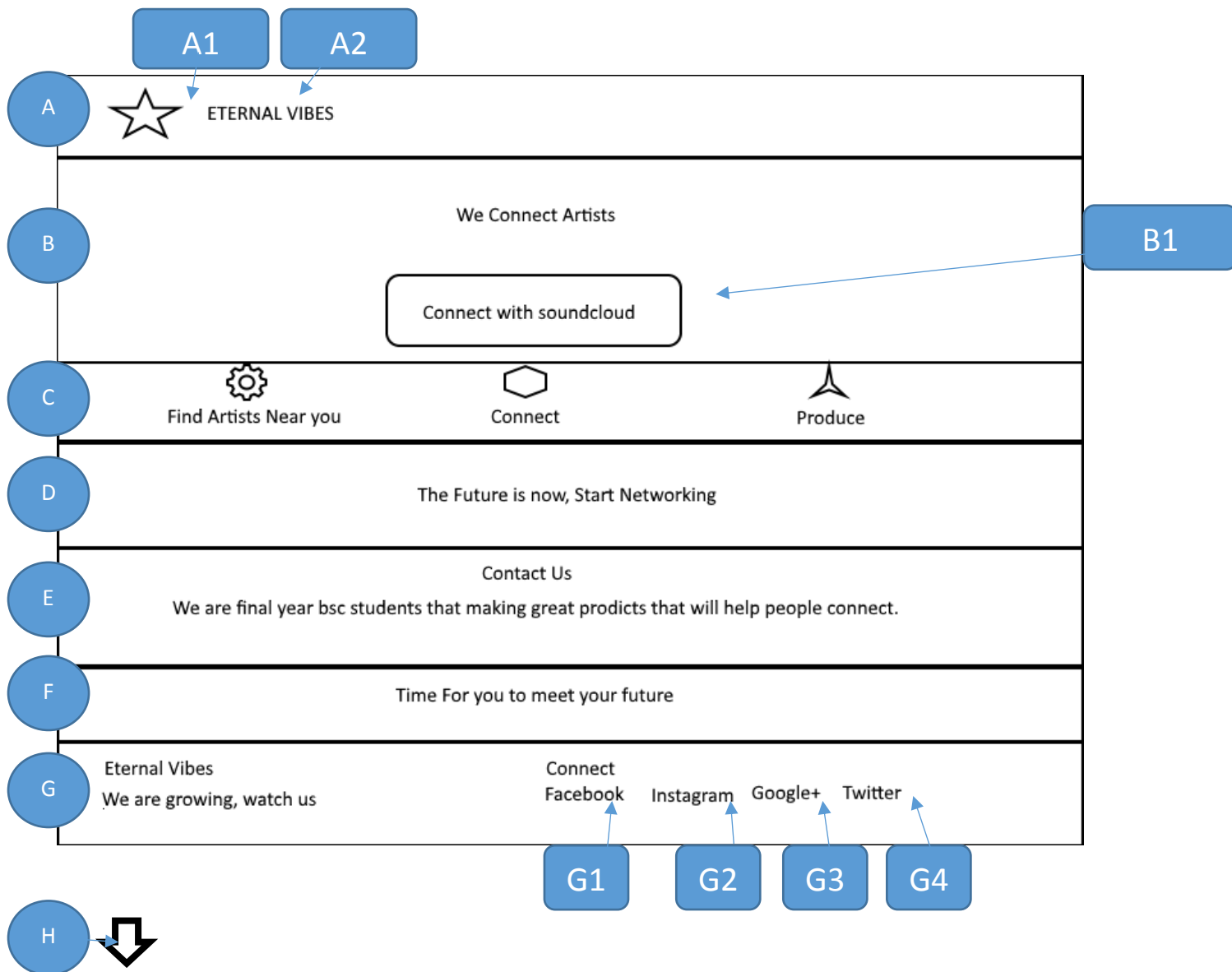
*Play Store App Page is an external page that we won't implement as our app won't be going into production due to the constraints of the Intellectual Property of the University. Conceptually this is a page that will allow the user to download the page.

*SoundCloud Login page is an external page that we use to authenticate users, as you need a SoundCloud account to access our site, SoundCloud handles the registration and Login process for SoundCloud and we just get the authenticated users information.

*Events and Posts are used interchangeably in this section to refer to a timeline entry; the context of the word will further clarify the Reference.

Landing Page

This is the first page of the website, the page that anyone interested in the website will arrive at first. It includes information about the features of the website, project and team. There are no restrictions on whom has access to this page, users can choose to engage with the website at this point or they can choose to not invest into it. The colour usage of the page follows the 60/30/10, of white/light brown/orange accordingly.



A: Navigation Bar: The left side of this bar stays the same throughout navigation. The right side has changes throughout navigation.

A1: Icon: The logo of our project, clicking this will refresh the current page (Landing Page).

A2: Label: System Name, our systems name, also the websites name, clicking this will refresh the current page (Landing Page).

B: Row: This is the first row of the site, it has a background image, title and Button.

B1: Button: Login via SoundCloud Button: Clicking this button will redirect you to the SoundCloud login system. After successful Login on SoundCloud, the user will be redirected to the Dashboard page in the original tab.

C: Row: This is the 2nd row of the page, it has information about the site, with icons supporting the below text, there is no interaction. No background Image.

D: Row: This is the 3rd row of the page, it has a background image and some overlaying text, no interaction.

E: Row: This is the 4th row of the page, it has a short bio of the group to market ourselves to people who want to contact us, no background image and no interaction.

F: Row: This is the 5th row of the page, it has a background image and some overlaying text, no interaction.

G: Footer: This is the footer of the page; it will have the same colour and content throughout all pages.

G1: Button: A Facebook button that links to our Facebook page.

G2: Button: A Instagram button that links to our Instagram page.

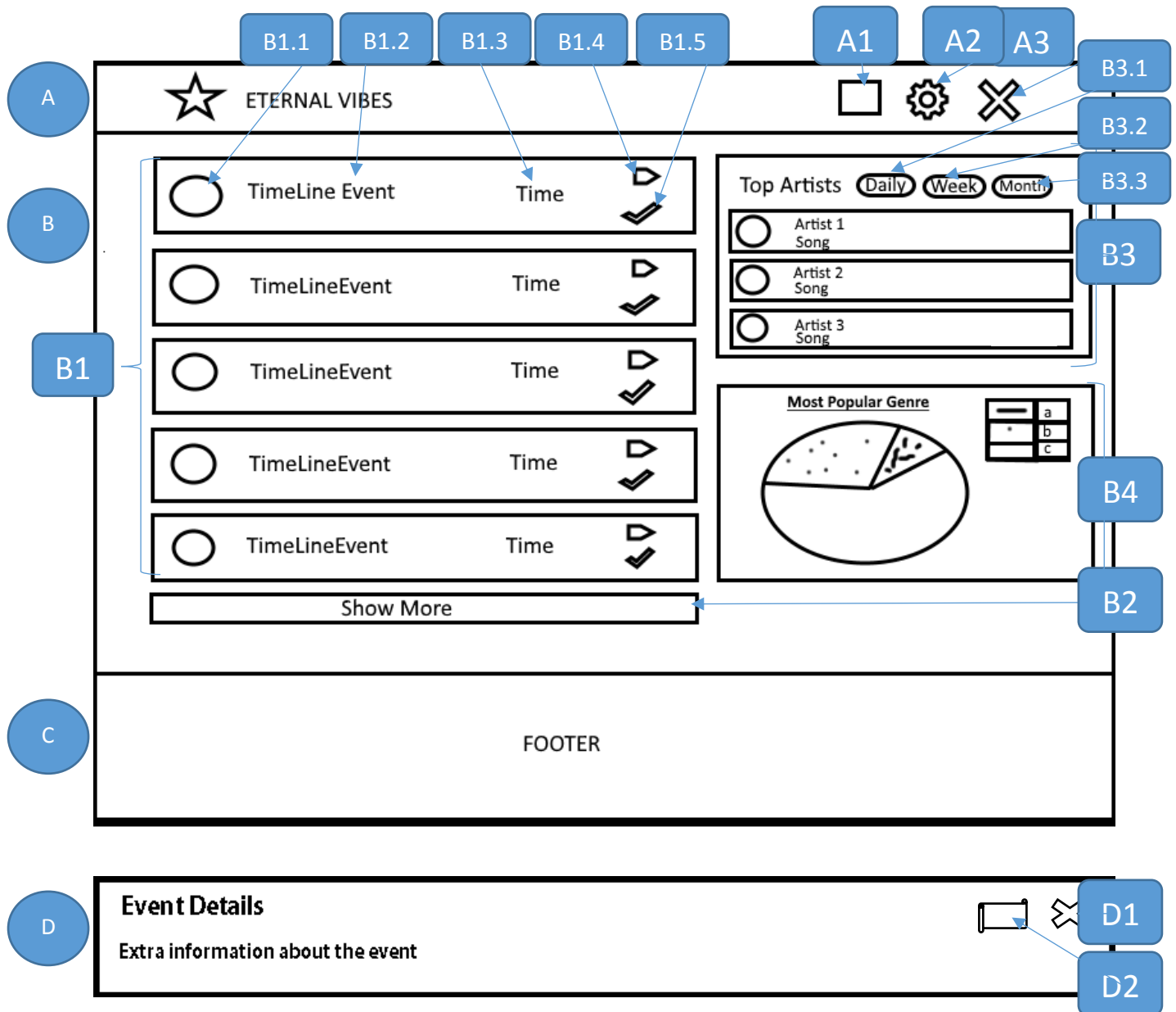
G3: Button: A Google+ button that links to our Google+ page.

G4: Button: A Twitter+ button that links to our Twitter Page.

H: Fixed Label: This label will be on a fixed position on the middle of the bottom edge of the view port. This label is there to indicate that there is more information remaining on the page. There is no interaction with this label. When the bottom of the view port is at the bottom of the footer, this icon will fade out and be replaced by nothing, it will not fade back in when you scroll back up.

Dashboard Page

Users that have successfully logged in via SoundCloud will be redirect to this page with information unique to them being displayed. This page is the main selling point of our website; users can view events that have happened amongst friends that they have made on the app. Statistics about the site can also be viewed and the calendar time ranges of the data collected can be chosen from a list of standard options. Only users who have successfully logged in via SoundCloud can view and use this page. It uses the same colour pallet as the Landing Page.



A: Navigation Bar: Same left side as on the Landing Page. The right side has actions that can only be performed while logged in.

A1: Button: The icon will be a refresh icon, this is the Refresh button, it will launch an ajax call that dynamically changed B1, B2 and B3 to their most recent data without page reload. In Summary, B1, B2 and B3 will be deleted and new data will be fetched and displayed.

A2: Button: This is the Settings button; it will Direct the user to the Settings Page on the same tab. Not done dynamically.

A3: Button: This is the logout button, it will end the user's session and log them out, then it will redirect them to the Landing page.

B: Section: This is the main div of the page that will show events in a timeline, top artists and the most popular genres.

B1: List: This is a list of events that have happened in the past 24 hours, sorted by most recent. 10 List items are displayed at a time.

B1.1: Image: The Profile Pic of the user that triggered the event, no interaction.

B1.2: Label: The Category of the event that happened, no interaction.

B1.3: Label: How long it has been since the event happened

B1.4: Button: This is the View More button which will show more detailed information about the event, the list item for which the view more button was clicked on will fade out and Section Labelled "D" will fade in. Displaying the type of event and more event detail.

B1.5: Button: This is the Finished with Event Button. When the button is clicked, it will remove the list item from the list and shift all other list items up, a new list item will then replace the empty space at the bottom of the list. This is all done dynamically and with smooth transitions.

B2: Button: This is the Load More List Items button; it will load the next 5 events that happened after the last list items timestamp. This List items will be added dynamically using ajax after the last list item in the original list. The Load More List Item button will then be shifted underneath the new last list item. This event can be repeated. If there are no more events to load, then the Load More button will change to a warning colour and be disabled so no more events can be fired off from it.

B3: List: This is the Top Artists List, this list has 3 list items, the items are the artists that have been the most active or who have got the most user "Upvotes" in the time that is specified by the 3 buttons above the list. This list doesn't automatically dynamically update after page load unless one of those 3 buttons are clicked. Each List Item contains the profile picture of the user, the name of the user and the Song they have picked to display on their profile.

B3.1: Button: This is the Sort by Daily button, it will sort the B3 List with the most upvoted artists on that day, the list will be dynamically updated using ajax with 3 new list items with the time range of a day.

B3.2: Button: This is the Sort by week button, it will sort the B3 List with the most upvoted artists on that day, the list will be dynamically updated using ajax with 3 new list items with the time range of a week.

B3.3: Button: This is the Sort by Month button, it will sort the B3 List with the most upvoted artists on that day, the list will be dynamically updated using ajax with 3 new list items with the time range of a Day.

B4: Graph: This is a pie chart that shows the different provisions of genres between every user with no date constraint. The Pie chart has a Title and a legend, hovering over the sections of the pie chart will show the value that is assigned to that section. There is no other interaction with these pie charts.

C: Footer: Same as Landing Page Footer

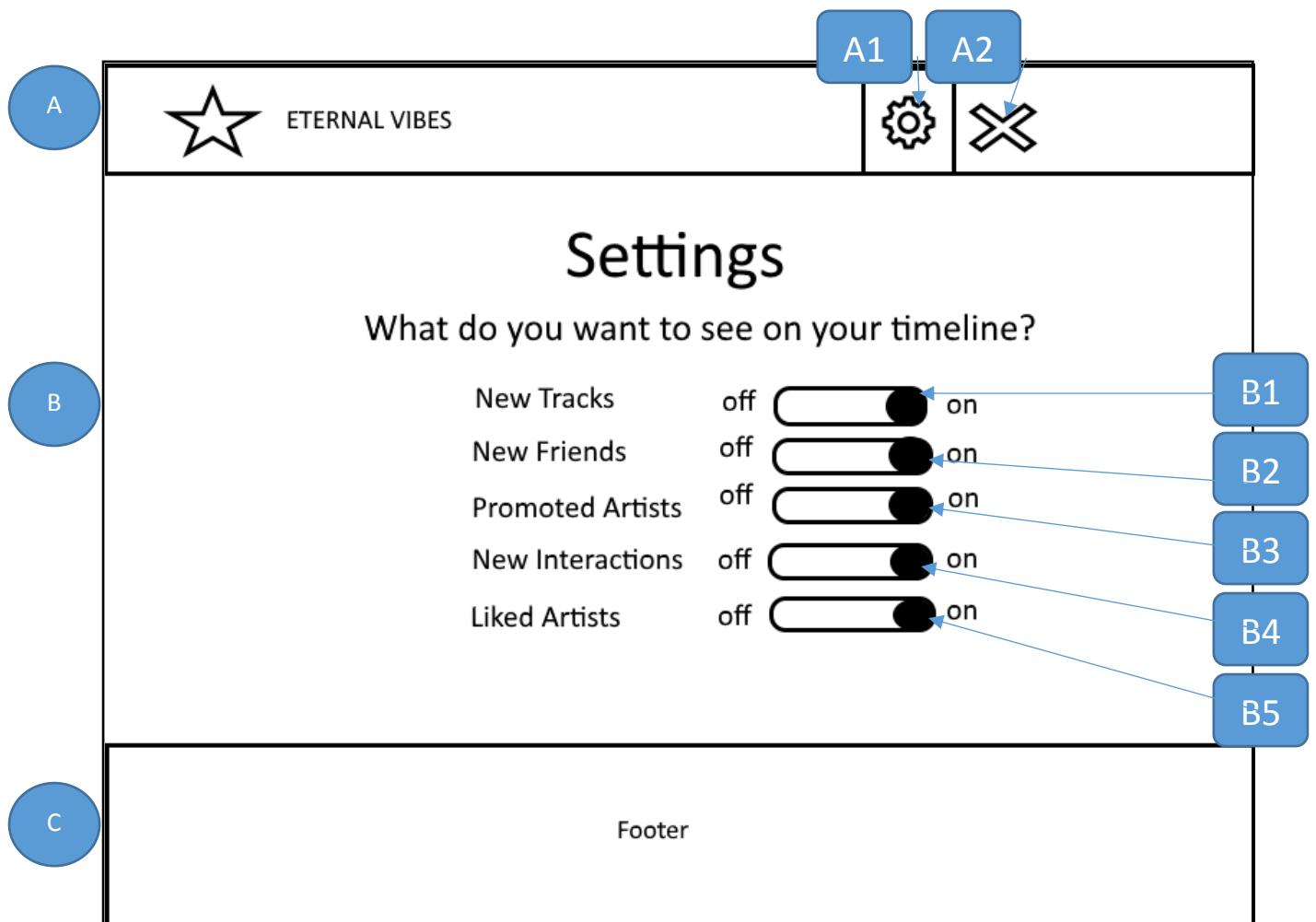
D: Row: This is a row that is displayed when B1.4 is clicked.

D1: Button: This is the close button; it will fade D out and replace it with the original display of the timeline event.

D2: Button: This is the flag post button; it will flag the selected timeline event for moderation by an admin for it being offensive or if it's spam. A tiny popup will also display in the top right corner of the viewport once the user has clicked the button in order to show that the post was flagged successfully. Flagged posts will either be removed or stay their depending on the admins decision.

Settings Page

This is the settings page for the users Dashboard Page, this allows the user to decide on what events will show up on their timeline. This is a form of timeline customization that will allow the user to only see information that they think is of value to them. The page is only accessible to a user that has successfully logged in via SoundCloud. The page will follow the same color pallet as the Landing Page.



A: Navigation bar: Left side is the same as the header on the Dashboard page. Right Side has been changed to fit the page

A1: Button: This is the same Settings Page button as the one on the dashboard page, except the background for it has been changed to show it as the active page

A2: Button: This is the same Logout button as shown on the Dashboard Page

B: Section: This is the Settings Section; it shows settings that allow you to turn on and off events that you want to appear on your timeline feed. This is a form of customization for the user.

B1: Lever: This turns on and off the event of new tracks appearing on your timeline. This will take effect on the next dashboard page load. The setting is set forever until changed by the user again.

B2: Lever: This turns on and off the event of new Friends appearing on your timeline. This will take effect on the next dashboard page load. The setting is set forever until changed by the user again.

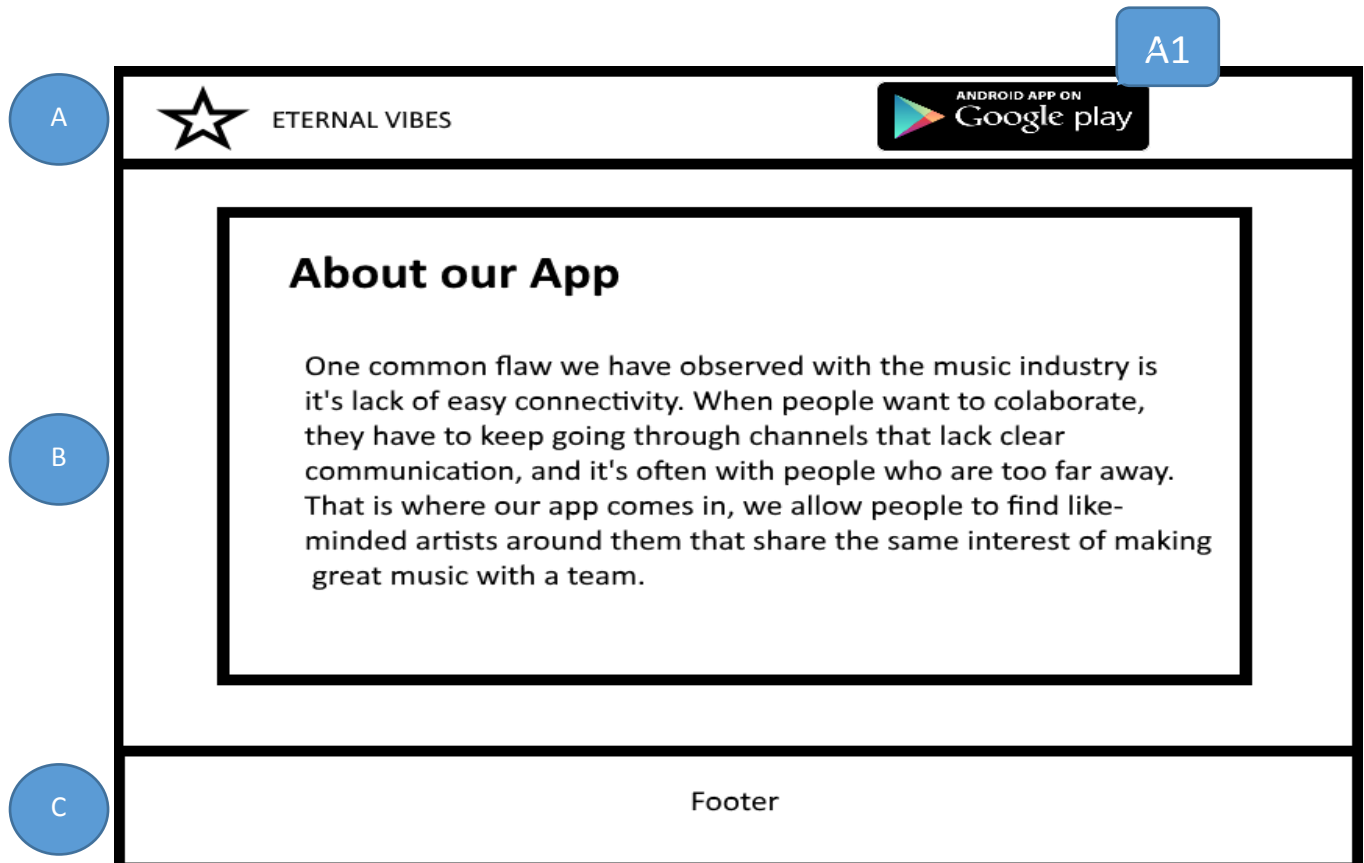
B3: Lever: This turns on and off the event of Promoted Artists appearing on your timeline. This will take effect on the next dashboard page load. The setting is set forever until changed by the user again.

B4: Lever: This turns on and off the event of new interactions between your friends appearing on your timeline. This will take effect on the next dashboard page load. The setting is set forever until changed by the user again.

54: Lever: This turns on and off the event of artists that you liked appearing on your timeline. This will take effect on the next dashboard page load. The setting is set forever until changed by the user again.

App Info Page

This page will have details about our Android Application, it will detail the importance of the application, why the user would use it and how to obtain it. It will also include a link to download the app, unfortunately that link will not work as putting the app on the store will violate the Intellectual Property of the University of Johannesburg for whom we are developing the project for. There are no restrictions on which users can view this page as it's a branch off the landing page, a page of information for the public with no real investment in the site. It will follow the same colour pallet as the Landing Page.



A: Navigation Bar: The Left side of the navigation bar is the same as the Navigation Bar on the Landing page. The right side is specific to this page.

A1: Button: This is the app store button; it has a graphic provided by google that allows users to be redirected in a new popup window to the Play Store web page for our android application when the button is clicked.

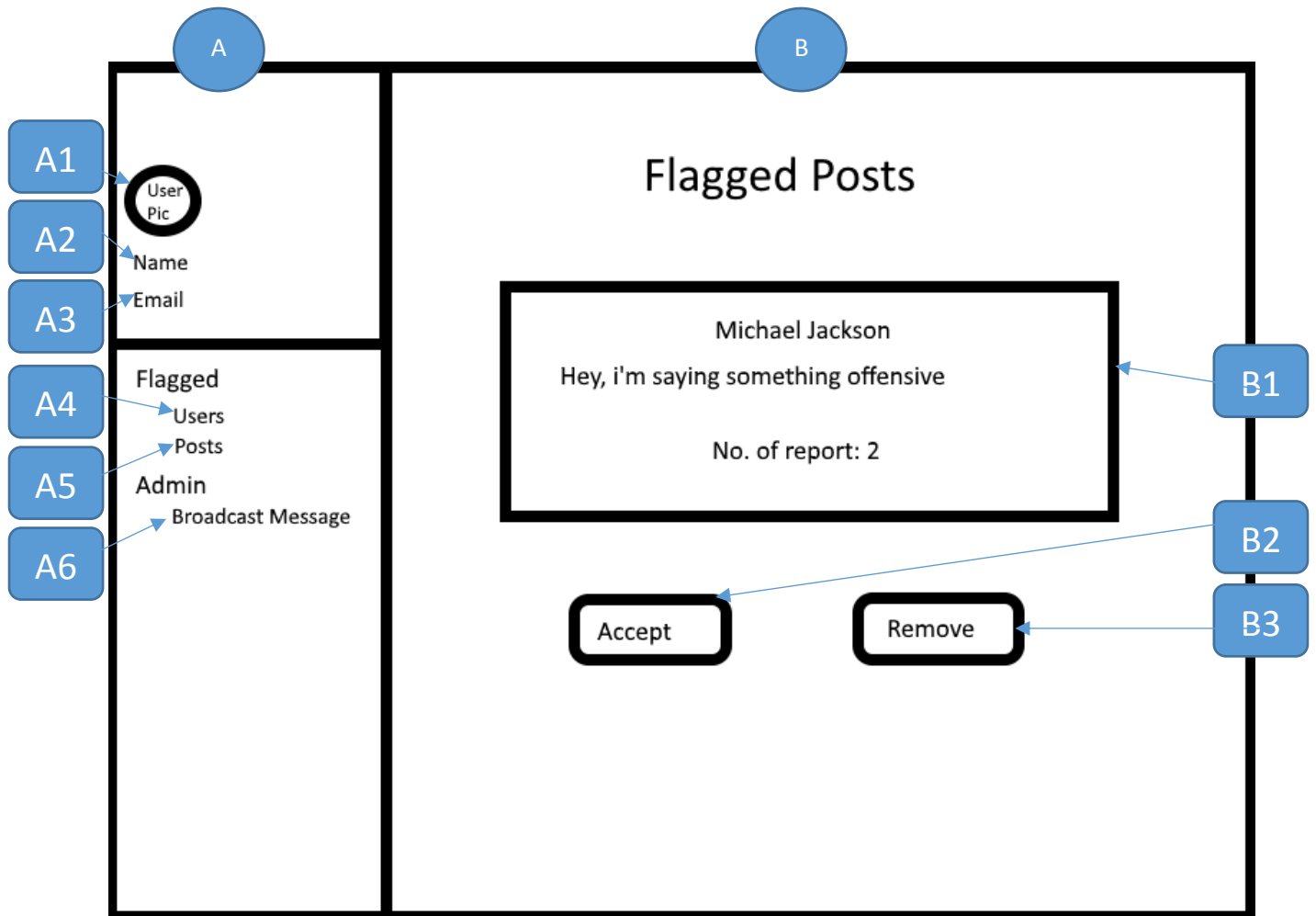
B: Section: The about our app section, this section has a card like interface that displays some quick information about why our app is important to artists and the music industry. There is no interaction.

C: Footer: This is the same footer as the Footer on the Landing Page.

Flagged Posts Page

This page is the default page that an administrator will be redirected to after successful login from SoundCloud. It then has the same weighting of importance as the other pages, it's just used more often so it's set to the default. This will be replaced by a Admin Dashboard page in future developments.

This page will allow an Admin to Remove or Accept post that have been flagged by users for moderation. The admin will accept a post if they feel there is no violation, if there is violation then they will remove the post, reasons for removal are if the event is considered spam or offensive then it will be removed. If the admin accepts the post then it will not be removed, if they click remove then it will be deleted from the system. This page is only accessible by Admins who have successfully logged in via SoundCloud. This page will follow the same colour pallet as the Landing Page.



A: Navigation Bar: The navigation has moved to the right side for the admin pages and statically stays present on the Flagged Users Page and the Broadcast Message page. The side navigation bar gives a more administration feeling to the pages and the admins identity is always displayed to ensure their identity.

A1: Image: A Rounded image of the admins profile pic.

A2: Label: The Name of the admin.

A3: Label: The email of the admin.

A4: Button: This button redirects the admin to the Flagged Users page in the same window, not dynamically, it's a new page load.

A5: Button: This button redirects the admin to the Flagged Posts page in the same window, not dynamically, it's a new page load.

A6: Button: This button redirects the admin to the Broadcast Message page in the same window, not dynamically, it's a new page load.

B: Section: This is the page specific section for Flagged Posts, it's a simple interface that allows admins to Review potentially malicious and offensive flagged posts.

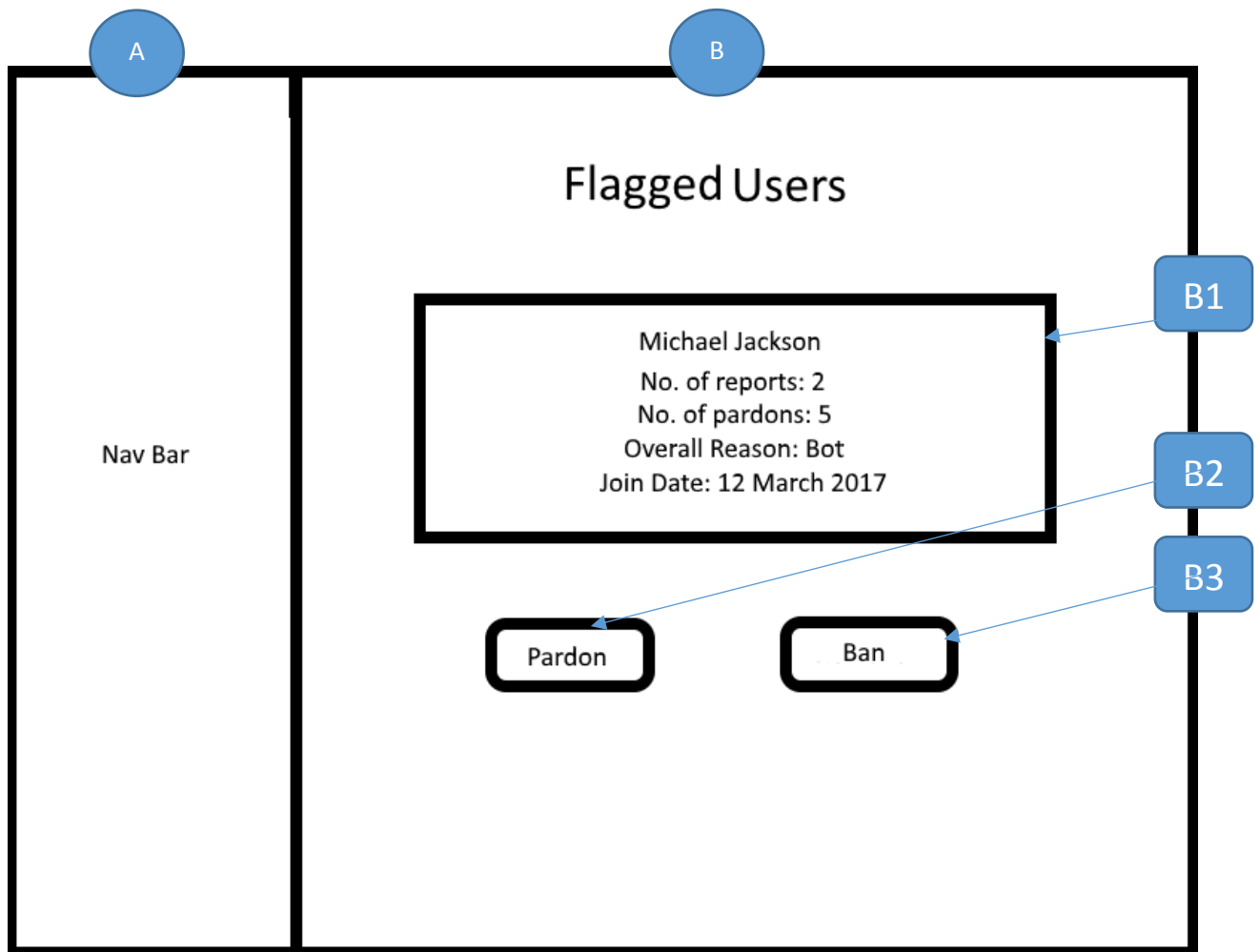
B1: Section: This section will have a card type look, it will contain the name of the artists that published the offensive post, the posts text and the amount of times that the post was flagged.

B2: Button: This is the accept post button, clicking this will leave the post, leaving it in place. It will then be removed by a transition that will move it off the right of the B section. Another B1 will then be dynamically transitioned from the Left of B section to the same place as the previous post. This ensures user friendly and fast transition of reports. If there are no more reports then a card will not transition from the left to the middle of the screen.

B3: Button: This is the remove post button, clicking this will remove the post from the Timeline that is was flagged on and all other occurrences of it. Clicking this button will have the same Transition effects as button B2.

Flagged Users Page

This page will allow an Admin to Pardon or Ban users that have been flagged by other users for moderation. The admin will Pardon a user if they feel there is no violation, if there is violation then they will ban the user, reasons for removal are if the user is a bot account or is offensive. If the admin pardons the user then they will not be banned and Pardon Counter in the database will be incremented, if they click ban then the user will be restricted on the system. This page is only accessible by Admins who have successfully logged in via SoundCloud. This page will follow the same colour pallet as the Landing Page.



A: Navigation bar: This is the same navigation bar as the Flagged Posts Page, the only difference is that button to reach this page will be set as active to change the CSS to indicate to the admin what page they are on.

B: Section: This is the page specific section for Flagged Users, it's a simple interface that allows admins to Review potentially malicious and offensive flagged users.

B1: Section: This section will have a card type look, it will contain the name of the artists that has been flagged, the number of times he has been flagged, the number of times they have

been pardoned, the overall reason as to why they were reported and the day that they first accessed the website or app.

B2: Button: This is the pardon user button, clicking this will pardon the user, leaving them in the system. It will then be removed by a transition that will move it off the right of the B section. Another users B1 section will then be dynamically transitioned from the Left of B section to the same place as the previous B1 section. This ensures user friendly and fast transition of reports. If there are no more reports then a card will not transition from the left to the middle of the screen.

B3: Button: This is the ban user button, clicking this will remove the user from the database and prevent them further access. Clicking this button will have the same Transition effects as button B2.

Broadcast Page

This page will allow an Admin to create an event that will be displayed on every user's timeline, alternatively the admin can specify which users they would like to broadcast these events to by specifying their UserID and separating them with a comma if there are multiple users. This page is only accessible by Admins who have successfully logged in via SoundCloud. This page will follow the same colour pallet as the Landing Page.

The wireframe illustrates the layout of the Broadcast Message page. It is divided into two main sections: a left sidebar and a main content area.

- Section A:** A circular callout labeled 'A' points to the 'Nav Bar' (Navigation Bar) on the left sidebar.
- Section B:** A circular callout labeled 'B' points to the 'Broadcast Message' title in the main content area.
- Form Fields:** A central form box contains the following elements:
 - Message:** A text input field with the placeholder text 'Our Server will be down tomorrow'. A callout labeled 'B1' points to this field.
 - Only to UsersIDs (Optional, comma seperated):** A text input field for specifying target users. A callout labeled 'B2' points to this field.
 - Submit:** A button to submit the broadcast message. A callout labeled 'B3' points to this button.

A: Navigation bar: This is the same navigation bar as the Flagged Posts Page, the only difference is that button to reach this page will be set as active to change the CSS to indicate to the admin what page they are on.

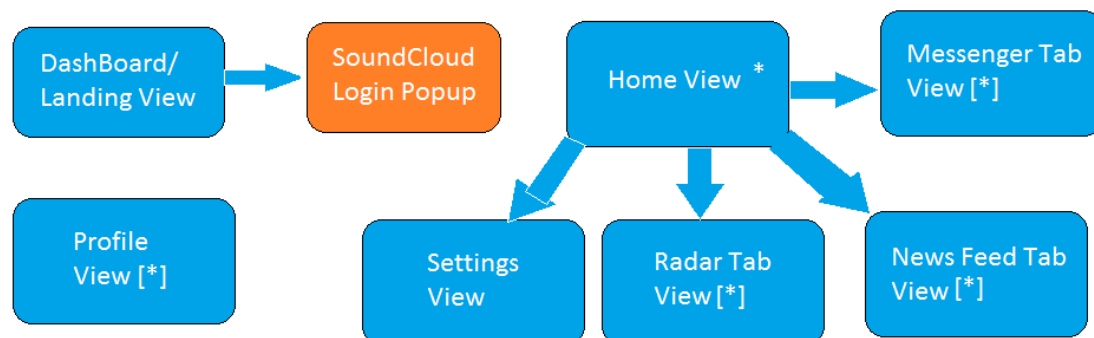
B: Section: This is the page specific section for Broadcast Message page, it's a simple interface that allows admins to broadcast messages to every user's timeline, unless specific users are specified.

B1: Input Box: This is the input box for the message that will be broadcasted to every user, it has a 120-character limit to making sure that a user's timeline is not flooded with broadcasted messages. This field is required.

B2: Input Box: This is the input box for specific comma separated UserIDs that will only receive the broadcasted message. This field can be left empty if the broadcast must go to everyone.

B3: Button: This is the submit button that will submit the message to the users specified or to everyone if not specified, it will be disabled if no characters are entered into the Message input box. This is to ensure that null messages aren't used to flood a user's timeline.

Mobile App Map

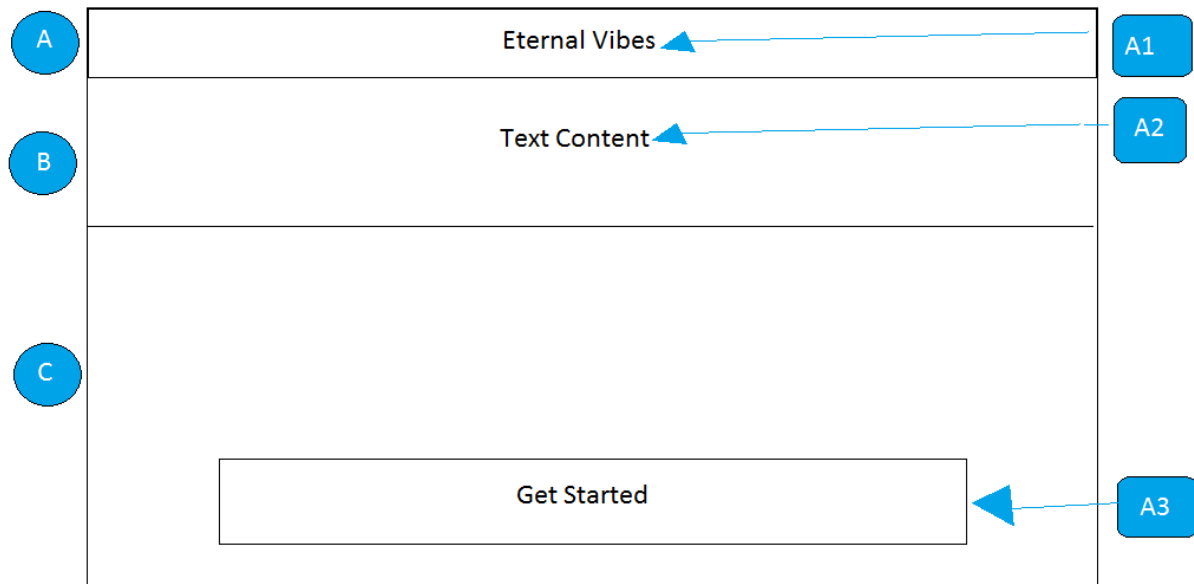


*The home view is an activity that will display one of 3 different tabs. Each tab is loaded depending which way the user swipes; the default tab that every user will see after viewing the Dashboard is the News Feed tab.

*To view a profile, users can select it from any of the three available tabs such as the News feed, Messenger and Radar and it allows users to perform various actions such as follow or message the user

Dashboard

This is the first page that the users see, they can view basic information and then proceed to an external Login Page.



A: Information Bar: This bar will contain text that displays the name of our project.

B: Text Content: This will display various content that the user will see when they first use the app.

C: Other Content: This will be a place to hold extra content for increased functionality such as buttons, labels and text views.

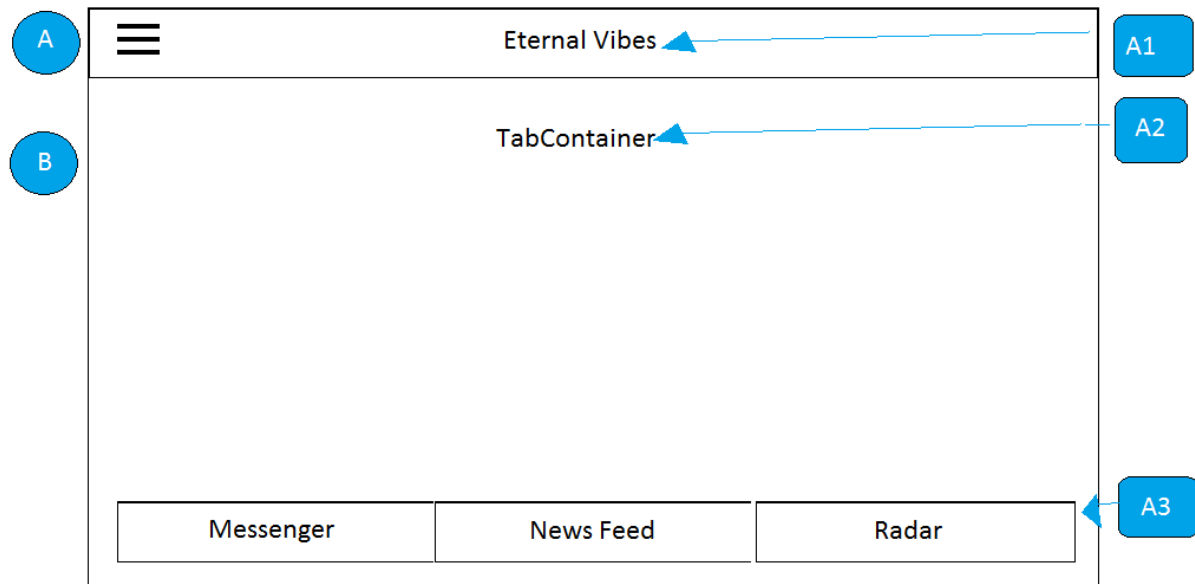
A1: Text Label: Information about our project name and this is also seen on Home View.

A2: Text: Content for basic information.

A3: Button: A button to allow the user to get started and to let the user log in.

Home View

Once a user has been logged in correctly and successfully, the user will be directed here where the user can view Multiple Views on one page, and access Options via Drawer.



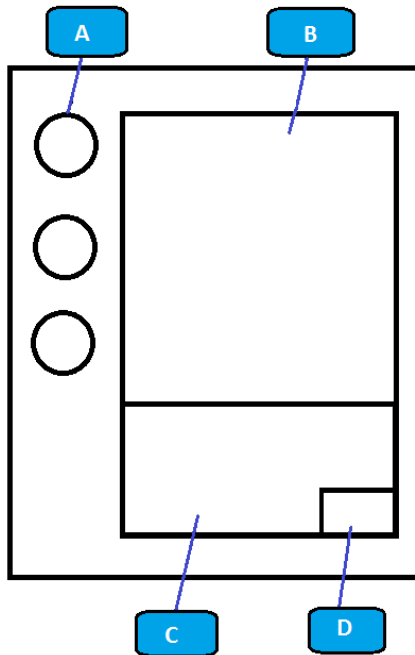
A1: Project Name: This is the name of our project that every user will see.

A2: TabContainer: This tab container will display various types of information depending which way the user swipes.

A3: Container: Shows the names of the tabs for the various tab sheets that will be displayed.

Messenger

We will be making use of a custom but messenger that allows users that have logged in to communicate with each other by sending text messages. This will make it convenient to schedule meetups as users find other artists.



A: Contact: The user will first select a contact which allows messages to be sent to this person.

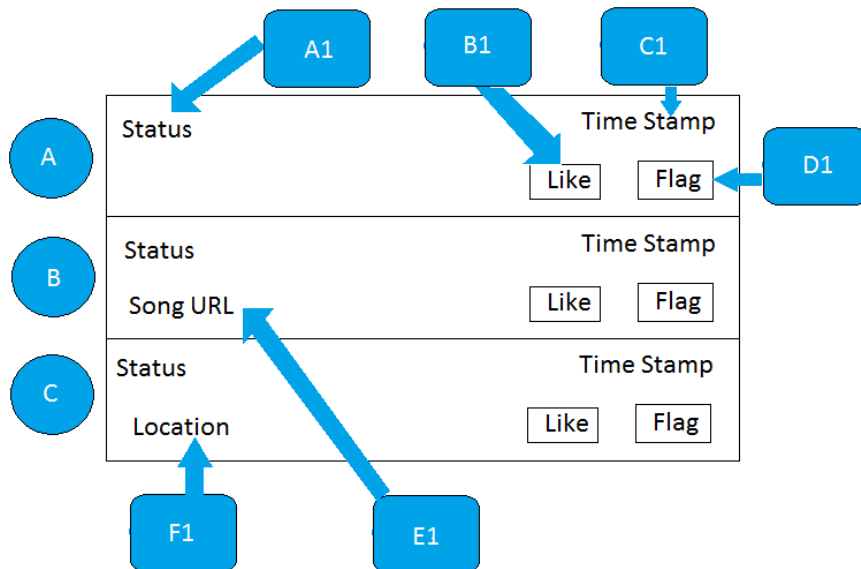
B: Text View: This will display a list of messages that the user sends and receives from other users.

C: Typing Area: This is the area that user will be presented with a digital keyboard and will be able to type a message.

D: Send Button: This will send the message to the selected contact in A.

Newsfeed

The Newsfeed will show a template list of related news within this tab. Users will be able to perform simple actions such as liking a post (other users can post a status) or flagging another user.



A: Normal Status or post

B: User uploaded a new Song snippet to SoundCloud

C: User Checked into a new Area

A1: Status Message: This message shows what state the user is in such as a flagged state, banned state or no state (which means this user has no issues.)

B1: Like Button: Users can Like The post

C1: Timestamp: This displays the time the Post was posted.

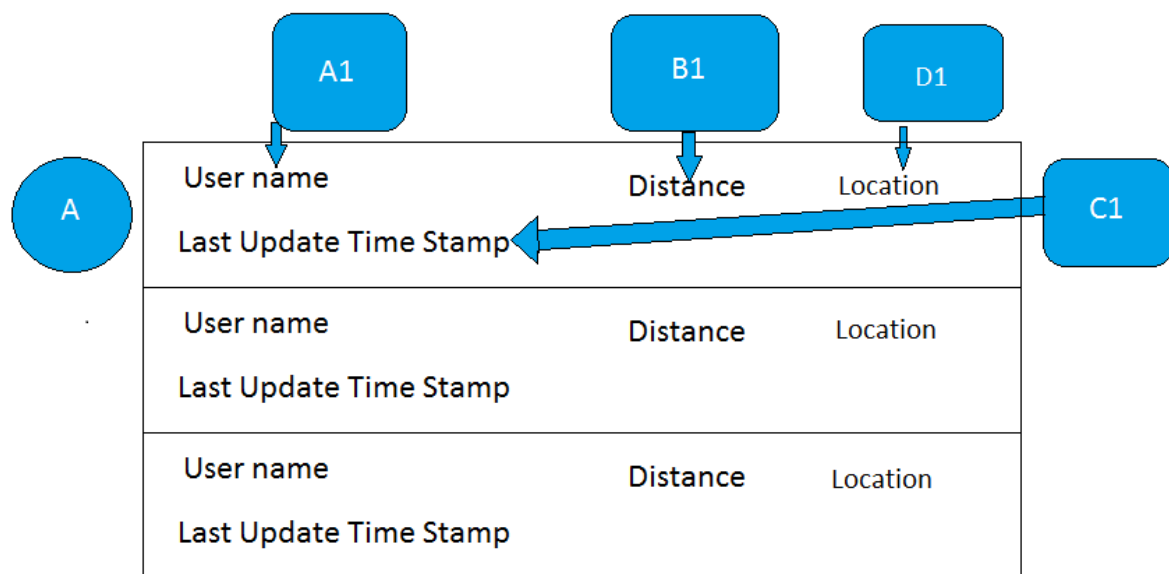
D1: Flag Button: Users can report and flag inappropriate posts.

E1: Song URL: to sound cloud where user can listen to the song or song snippet

F1: Location: The last location the user was in.

Radar

Displays a list of users nearby to user of the app, a User can touch and view profile of the selected user and view information about them, send a message or follow them.



A: Template Layout: The layout of this interface is repeated for each user that is found and displayed,

A1: User name: This shows the name of the user found.

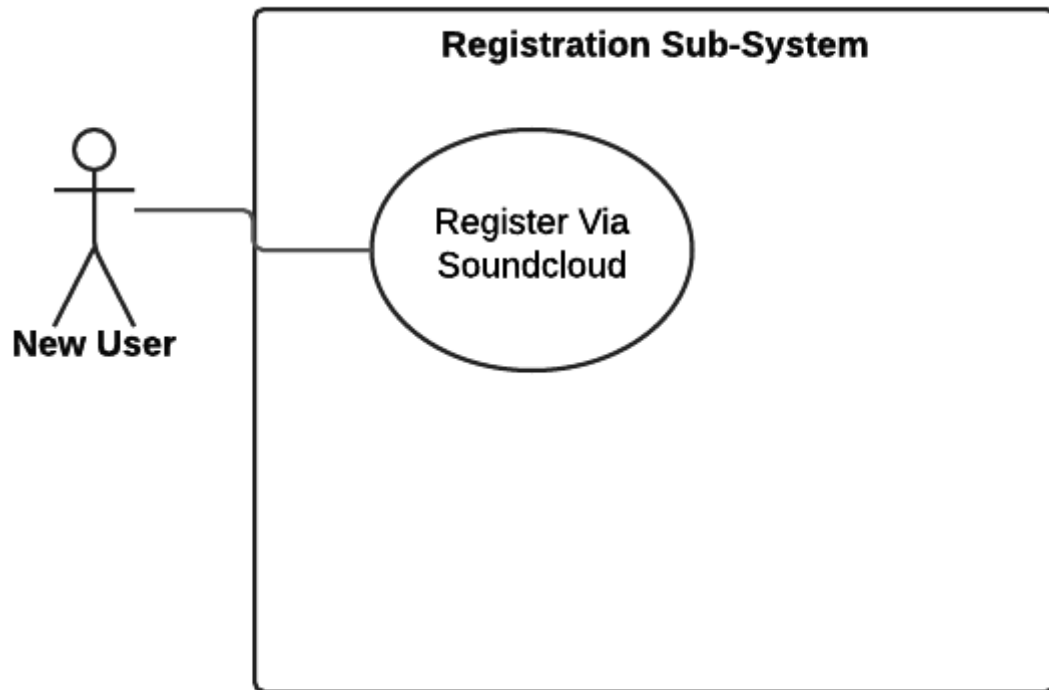
B1: Distance: Shows the distance in (km) between the users that were found and the current user's location.

C1: Time stamp: This indicates the last time the user's location was updated.

D1: Location: The general location of the user(s) found is displayed here such as Sandton.

Appendix

Updated Use Case Diagram



Use case: Register User Via SoundCloud

- 1) User registers using existing SoundCloud account
- 2) System requests account authorization using SoundCloud
- 3) System authorizes SoundCloud account
- 4) System Links account to User
- 5) User Fills in additional registration information
- 6) System creates account

Extensions

1a) User data does not have existing SoundCloud account

1. Redirect user to SoundCloud registration page

1b) User already exists on the system

1. System notifies user
2. User goes to login page

2a) System cannot request authorization User's SoundCloud account

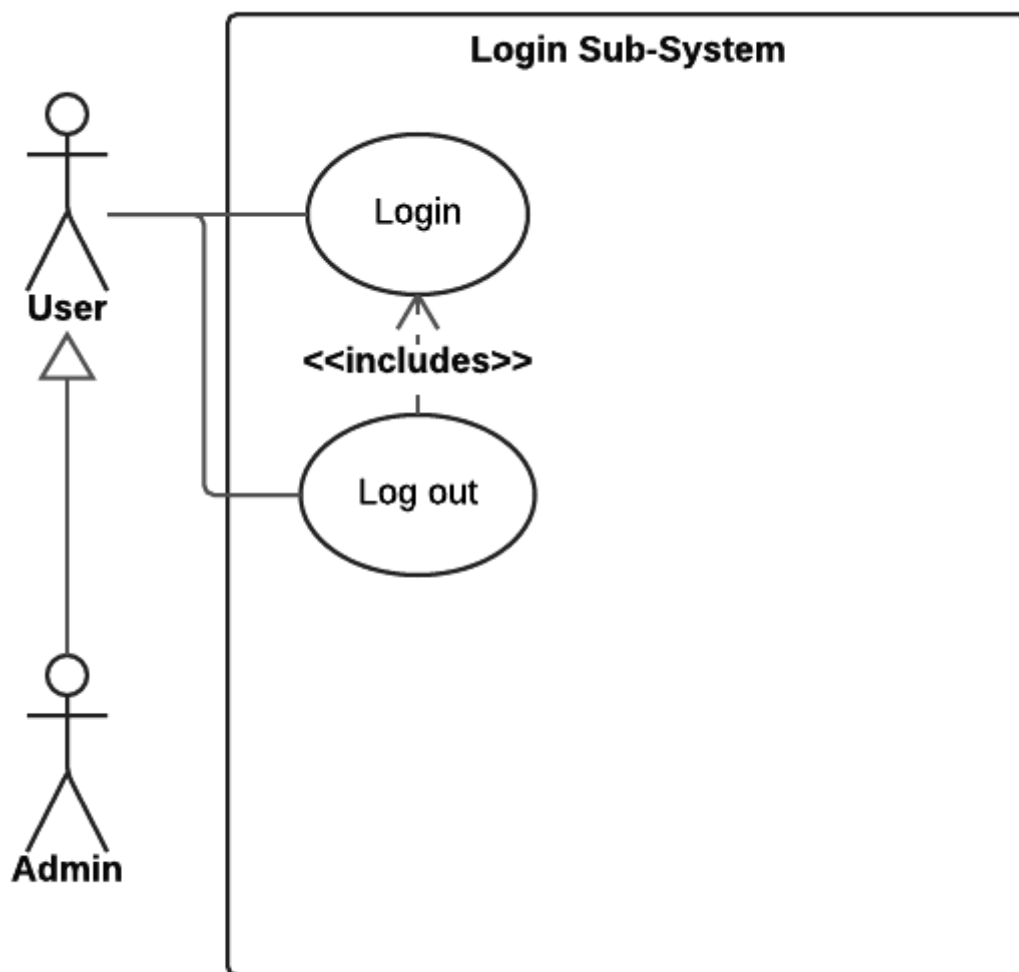
1. System notifies User
2. User may re-enter SoundCloud credentials

3a) Authorization fails (authorization timeout)

1. System notifies User
2. User tries again

4a) User inputs incorrect information

1. System notifies user
2. User re-enters information correctly



Use Case: Login

1. Users/Admin Enters Login Credentials
2. User i/Admin Logged into System
3. System is redirected to User is/Admin dashboard

Extensions

1a) User's/Admin's Login credentials are incorrect

1. System notifies User/Admin
2. User/admin re-enters credentials

1b) User/Admin account does not exist

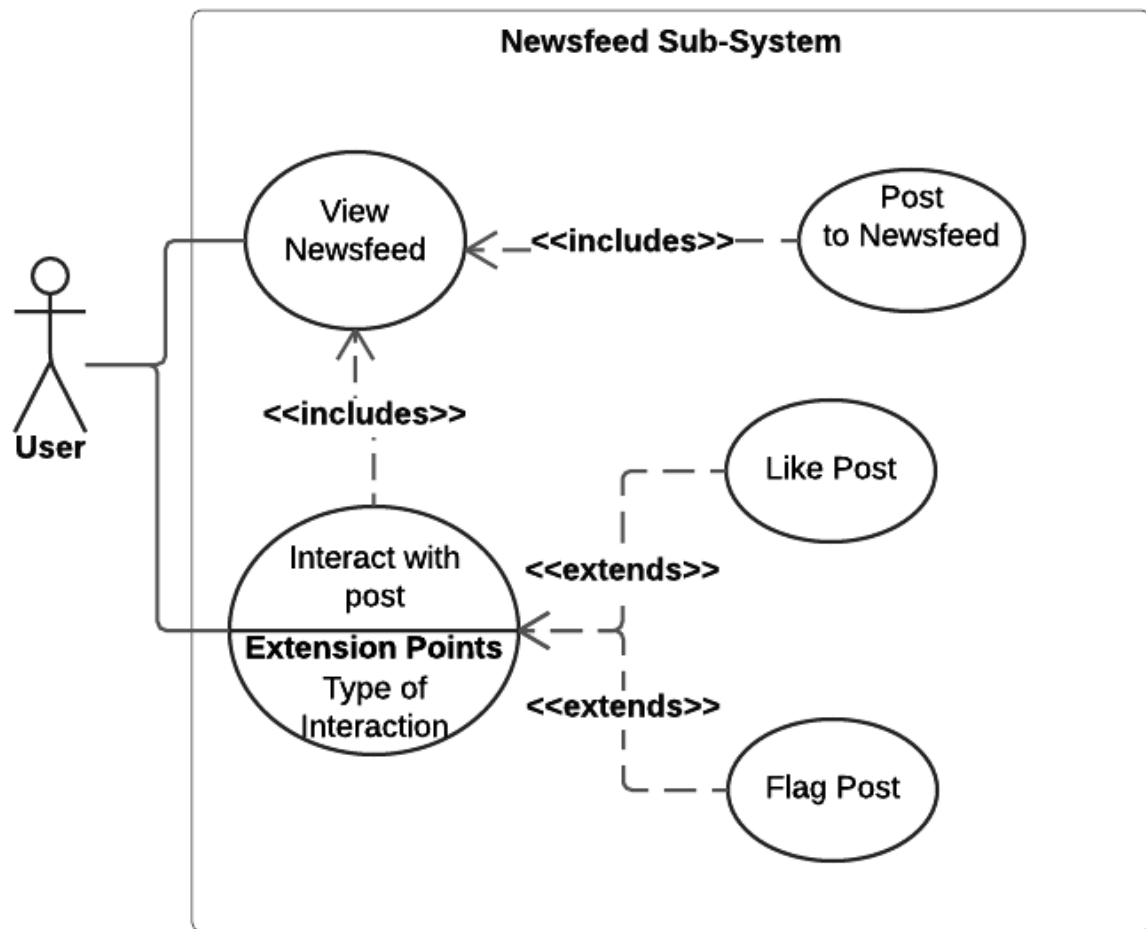
1. Notify User/Admin
2. User/Admin goes to registration page

1c) User/Admin forgets login credentials

1. User/Admin enters email address linked to account
2. System sends an email to User/Admin with their Login credentials
3. User/Admin tries again

Use Case: Logout

1. User/Admin is finished using the system
2. User Logs out of System
3. System redirects User/Admin to login/registration page
4. User closes system



Use Case: View Newsfeed

1. User goes to Newsfeed
2. System loads Newsfeed
3. User browses Newsfeed

Extension

2a) No internet connection

1. System waits for internet connection
2. System loads newsfeed

Use case: Post to Newsfeed

1. User goes to Newsfeed
2. System displays Newsfeed
3. User browses Newsfeed
4. User adds a post
5. System uploads post
6. User's Newsfeed is updated with new posts

Extension

1a) User discards post

1. System discards post

5a) No internet connection

1. Post not uploaded
2. System waits for internet connection
3. System uploads post

Use Case; Interact with post

1. User browses Newsfeed
2. User views an Artist's post
3. User Interacts with post

Use Case: Like Post

1. User browses Newsfeed
2. User views an Artist's post
3. User Interacts with post
4. User Likes/Flags post

Extension

4a) User Likes Post

1. System notifies Artist that their post has been liked

4b) User Flags Post

1. System notifies Admin on flagged post

1. User chooses Artist to message
2. System displays chat between the Users
3. User types a message
4. User sends message
5. System notifies Artist that new message received

Extensions

3a) Message discarded

1. User leaves chat
2. System discards message

4a) No internet connection

1. System notifies user
2. System waits for internet connection
3. System sends message when connection is established

Use case : Receive a message

1. Server sends a notification to User
2. User opens chat with Artist
3. User reads received message
4. User responds to message

Extension

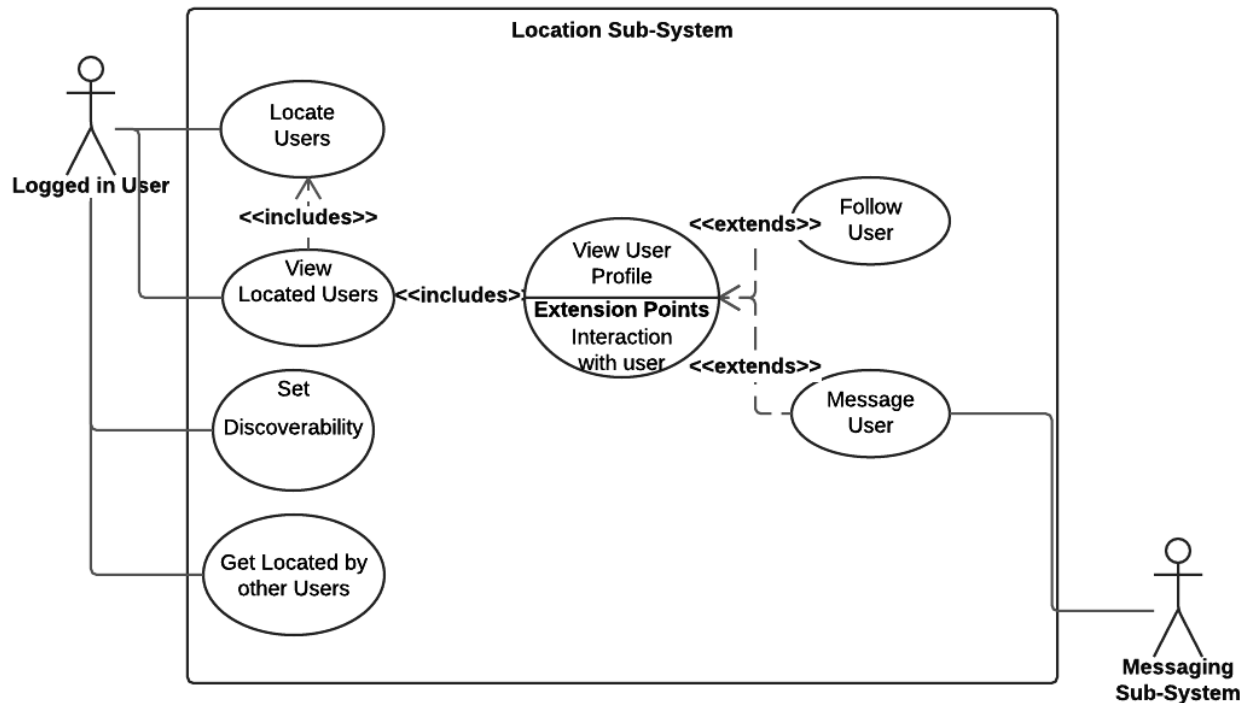
1a) No Internet connection

1. System waits for a connection
2. System sends notification once connection is established

4) User does not respond message

Use Case: Notify User

1. System notifies User that a message has been received
2. System notifies user based on internet connection



Use Case: Set discoverability

1. User sets discoverability on or off
2. System updates User's discoverability status
3. System stores User's location

Extension

2a) No internet connection established

1. System waits for internet connection
2. System updates User's discoverability status once internet connection established

Use Case: Locate User

1. User sets discoverability on
2. System updates User's discoverability status
3. System stores User's location
4. System locates nearby Artists
5. System displays located Artists
6. User views located Artists

Use Case: View User Profile

1. System locates nearby Artists
2. System displays located Artists
3. User views located Artists
4. User views a Artist's profile

Use Case: Follow User

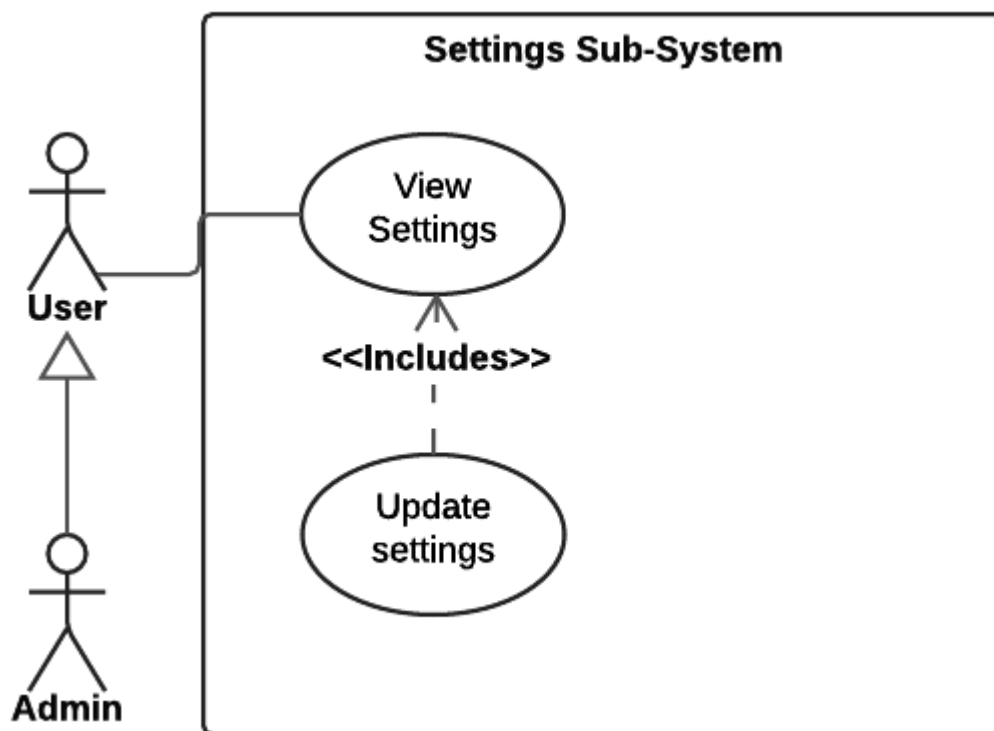
1. System locates nearby Artists
2. System displays located Artists
3. User views located Artists
4. User views a Artist's profile
5. User Follows Artist
6. System updates User's following
7. System updates Artist's Followers
8. System updates User's newsfeed
9. System notifies Artist

Use Case: Message User

1. System locates nearby Artists
2. System locates nearby Artists
3. System displays located Artists
4. User views located Artists
5. User views a Artist's profile
6. User sends message to Artist
7. System notifies Artist
8. System stores conversation

Use Case; Get Located by other Users

1. User's discovery is set on
2. System updates User's discoverability status
3. System stores User's location
4. Artists locate User
5. Artists interact with user
6. System notifies User

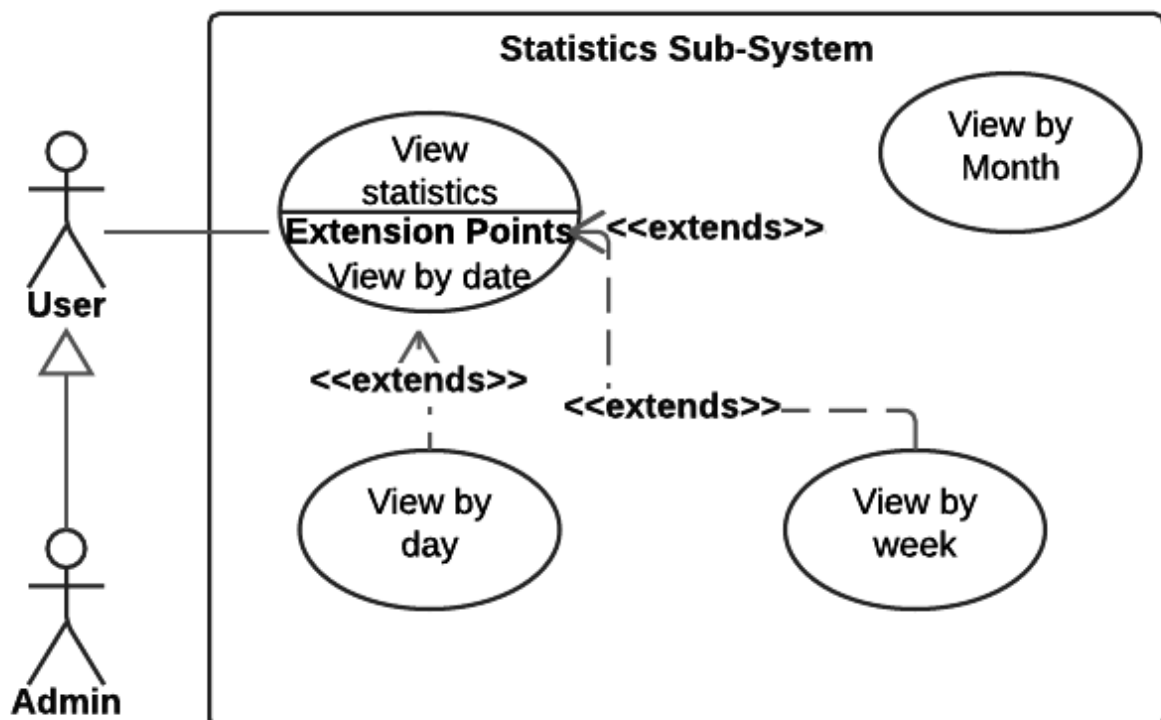


Use Case: View settings

1. User opens settings
2. System displays settings

Use Case: Update settings

1. User opens settings
2. System displays settings
3. System validates settings
4. System saves settings



Use Case: View statistics

1. System generates statistics
2. User/Admin views statistics

Use case: View by day

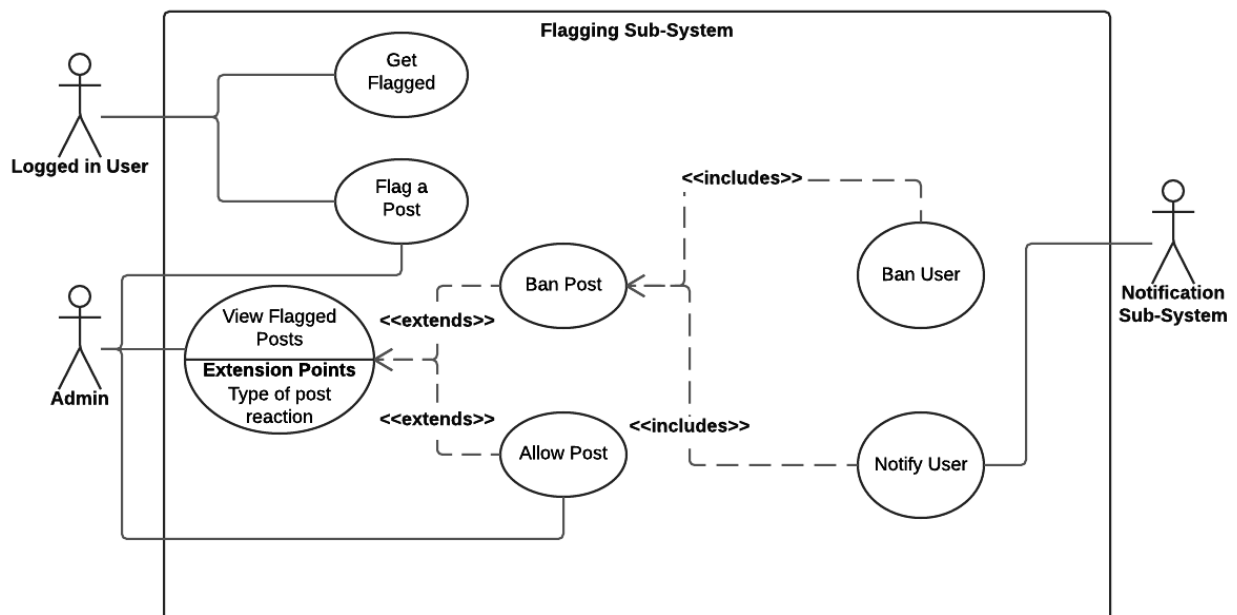
1. System generates statistics
2. User/Admin views statistics
3. User /Admin requests statistics view by day
4. System displays statistics by day

Use case: View by week

1. System generates statistics
2. User/Admin views statistics
3. User /Admin requests statistics view by week
4. System displays statistics by week

Use case: View by month

1. System generates statistics
2. User/Admin views statistics
3. User /Admin requests statistics view by month
4. System displays statistics by month



Use Case: Flag a post

1. User/Admin view an Artist post
2. User/Admin Flags post
3. System updates flagged posts
4. System updates how many times post has been flagged
5. System notifies Admin

Extension

2a) Cancel post flag

1. System updates flagged posts
2. System updates how many times post has been flagged

Use Case: Get Flagged

1. System flags User's post based on other Artist's opinion
2. System stores post as flagged
3. System notifies Admin

Use case: View flagged Posts

1. Admin selects flagged posts
2. System displays all flagged posts

Extensions

2. No flagged posts available

Use Case: Ban Post

1. Admin selects flagged posts
2. System displays all flagged posts
3. Admin Bans Post

4. System Removes Post from User's newsfeed and Artist's who follow User
5. System notifies user
6. System updates flagged posts

Extension

3a) Admin cancels ban

1. System keeps post as flagged

Use Case: Ban User

1. Admin selects flagged posts
2. System displays all flagged posts
3. Admin Bans Post
4. System Removes Post from User's newsfeed and Artists who follow User
5. System notifies user
6. System updates flagged posts
7. Admin Bans User
8. System notifies user on Ban

Extensions

6a) Admin cancels ban

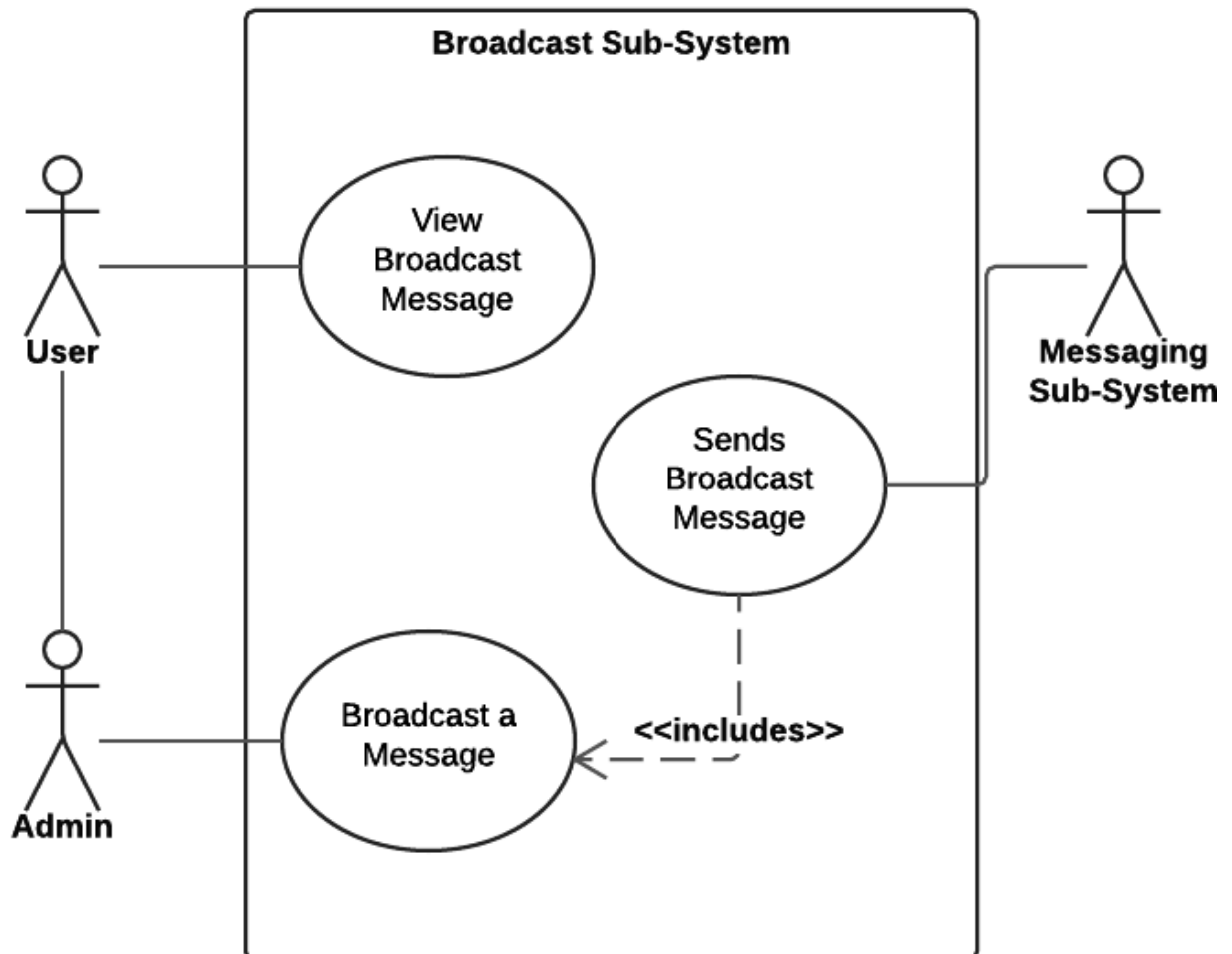
Use Case: Allow Post

1. Admin selects flagged posts
2. System displays all flagged posts
3. Admin approves post and removes flag
4. System keeps post on User's newsfeed and Artists who follow User
5. System updates flagged posts

Extensions

3a) Admin cancels approval

1. System keeps post as flagged



Use Case: View Broadcast Message

1. System sends broadcast to User
2. System notifies User
3. User views broadcast

Use Case: Broadcast a message

1. Admin sends broadcasts a message to list of Users

2. System sends message to list of Users

Extension

1a) Cancel broadcast

1. System discards broadcast

Use Case: Send Broadcast Message

1. Admin sends broadcasts a message to list of Users
2. System sends message to list of Users
3. System notifies Users