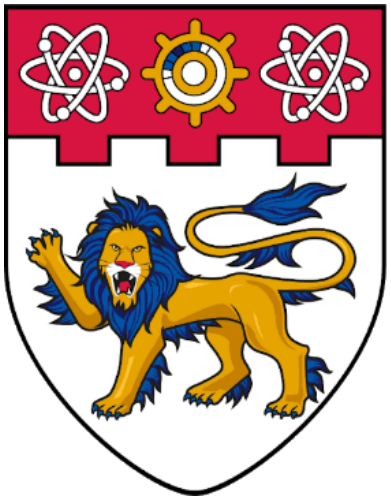

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ3002 Advanced Software Engineering

System Requirement Specification (SRS)

Lab Group: SSP4

Team Name: Inspiration

Royce Ang Jia Jie (Leader)
Manav Arora (Vice-Leader)
Jovan Huang Tian Chun
Clarence Hong Shi Man
Zhu Weiji
Tan Hui Zhan

1 Table of Contents

1 Table of Contents **1**

2 Problem Statement **3**

3 Overview **3**

3.1 Background 3

3.2 Overall Description 3

4 Investigation & Analysis Methodology **4**

4.1 System Investigation 4

4.2 Analysis Methodology 7

4.2.1 Feasibility study and requirements elicitation 7

4.2.2 System analysis and requirements specification 8

4.2.2.1 Perform an analysis of the problem using object-oriented techniques 8

4.2.2.2 Scope and Limitations 9

4.2.3 Object-oriented design using UML 10

4.2.4 Prototype **11**

5 Constraints **12**

5.1 Proprietary Hardware and Software 12

5.2 Project Schedule 13

6 Operational Requirements **14**

6.1 Request Support via Email 14

6.2 Application Services and Technical support 14

6.3 Administration Features 14

6.4 System failure and routine back up 15

6.5 Audit Trail 15

7 Functional Requirements **16**

7.1 Register 16

7.2 Login 16

7.3 Logout 16

7.4 Recommend Friends 16

7.5 View Recommended Profiles 16

7.6 View Personal Profile 16

7.7 View Friends Matched History 16

7.8 Edit Personal Profile 16

8 Input Requirements **17**

8.1 Email Address and Password 17
8.2 About Me, Interests and Hall 17
8.3 Query Keywords 17
8.4 Preferences for Drinks 17

9 Process Requirements 17
9.1 Backend Service Transactions via HTTP 17
9.2 Data Integrity 17
9.3 Data Validation 17
9.4 Performance 18
9.2 HTTP Transaction 18
9.2 Data Integrity 18
9.3 Data Validation 18
9.4 Performance 18

10 Output Requirements 18
10.1 List of Recommended Friends 18
10.2 Personal Profile and History of Matched Friends 18
10.3 Potential Friends Profile 18
10.1 List of Queried Drinks 19
10.2 Relevant details of drinks 19
10.3 User profile Details 19

11 Hardware Requirements 19
11.1 Client Computers and Mobile Devices 19
11.2 Network 19

12 Software Requirements 19
12.1 Client Operating Systems 19
12.2 Network system 19
12.3 Licenses 20

13 Deployment Requirements 21

2 Problem Statement

The COVID-19 pandemic has undoubtedly affected many students' academic and social experiences in their university journies. Therefore, we concluded that there is a need for a platform for Nanyang Technological University students to help them in finding new potential friends. We propose the development of 'Friendstagram', a personal friend recommender web app which is catered specifically for Nanyang Technological University students, to help them connect with new potential like-minded friends based on their interests and hall accommodation, so as to enhance their university lives.

3 Overview

3.1 Background

The COVID-19 pandemic has disrupted university students' daily lives. Many social restrictions have been put in place to prevent the spread of the virus amongst student cohorts. Co-curricular activities have been cancelled and lessons have been largely moved online, thus, reducing social interactions between students.

The inability to interact with others has caused students to achieve a less fulfilling university life. In a late 2020 survey conducted on 400 university students found that the shifting of in-person classes to online learning had caused a significant drop in students' satisfaction with the university courses. Students also felt like they do not belong to a community. Furthermore, when compared to the survey results conducted in the previous year, there was a 23% decline in "feeling engaged in coursework" and a 28% decline in "interacting with peers in the classroom". Being together with friends and in-person classes were also selected as the top 2 reasons why students valued being on campus. In Singapore, a survey conducted on the impact of the pandemic with 1,066 youths aged between 18 and 35 found that 58% responded they had become more fearful and 54% had responded they had become less sociable.

Therefore, we concluded that there is a need for a platform for Nanyang Technological University students to help them in finding new potential friends.

3.2 Overall Description

Our web application, 'Friendstagram', is a friend finder application catered specifically for students of NTU and it consists of 3 main features.

Firstly, users can register for an account and create a personal profile which best describes them. In the

profile, they can configure their hall residence, interests, contact information, etc. Secondly, through the profile that was set up, users may use the application to search for potential friends. The algorithm that we had designed will match other users with similar interests or stay in the vicinity and display their profiles. Lastly, users may then view the recommended friends and contact them.

The purpose of the web application is to create a social platform that allows NTU students who are having trouble socializing due to the COVID-19 social restrictions to strike up new friendships with like-minded peers in NTU. With a personalized profile, users can be recommended to friends whom they can easily connect with and befriend due to common interests and/or topics to talk about. These features could help to achieve our objective of creating a platform to enrich NTU students' university journeys.

4 Investigation & Analysis Methodology

4.1 System Investigation

The core functionality that the Friendstagram application shall provide is recommending potential friends to users. The system comprises a frontend client providing an interactable user interface and a backend system to process incoming requests. The account, profile, and matched friends history information are stored in the backend system. The developed algorithm will match users as potential friends. When the system receives a find friends request, the algorithm will match users with common interests and profile characteristics as potential friends. Consequently, the system will feedback the potential friends back to the requesting user.

4.2 Analysis Methodology

4.2.1 Feasibility study and requirements elicitation

The project team will consist of members who are proficient in web application development, quality assurance and project management. Prior to embarking on the design and development of the system, the project team will elicit the requirements by conducting interviews and surveys with existing NTU students to understand their pain points and frustrations better. The necessary system functionalities to address the pain points will then be defined. Subsequently, to ensure the feasibility of the system, the project team will further seek feedback from our target users, NTU students, on whether the functionalities would address their

frustrations.

To ensure the success of the project, a Risk Assessment study will also be conducted to identify, analyze and draft appropriate responses to each corresponding risk. Both positive and negative risks will be studied. The Assessment will be available in a separate document, *Risk Assessment study*.

4.2.2 System analysis and requirements specification

4.2.2.1 Perform an analysis of the problem using object-oriented techniques

Friendstagram will be analyzed and developed using an Object-Oriented Methodology. We will then use Unified Modeling Language to model the applications' features, use cases, structure and behaviors. We will be using this System Requirement Specification as a form of documentation for the project. The important features of the recommendation platform include:

- The ability to recommend relevant people based on users' preferences
- Allow users to review and browse through the recommended profiles
- The ability to show users the past recommendations
- Allow users to edit their profile and preferences

4.2.2.2 Scope and Limitations

Analytical methods include business analysis, requirements analysis, data analysis, process analysis and application architecture.

Business Analysis
Consultation with business analysts can provide advice on how to negotiate and communicate with sponsors. There will also be a discussion on the budget estimates, business plans and procedures, and project duration estimates.

Requirement Analysis
The development team will frequently contact system users to determine specific functional expectations, resolve conflict resolution and remove ambiguity in requirements as needed by different users and user groups. Documentation for all aspects of the project development process from start to finish is included. Also, we will avoid feature creep by setting expectations right at the start with the stakeholders. Apart from that, our end product needs to meet the needs of end-users. End-users' needs should not be adapted to fit the end product.

Data Analysis

After each recommendation, the data analysis team collects survey results generated by users. We then examine the data, clean it up, transform it, model it, draw relevant and useful information, and draw appropriate conclusions.

Process Analysis
Quality Control performs process analysis to ensure that traditional errors are avoided. Typical mistakes like this include lack of upstream activity and quality assurance, inadequate control, premature or frequent convergence, omission of tasks required for estimation, planning to catch up later, inefficient/incomplete code, overly optimistic schedule, poor risk management, contractor failure, poor planning, plan cancellation under pressure, time due to ambiguous front end and improper design. The team also enables the development team to follow the quality plan and proceed according to the plan without complexity or setbacks.

Application architecture
We will model the application architecture based on the different application features using UML. For our front-end, React and Redux frameworks will be used as they are component-based, efficient and have a smooth learning curve. For our back-end, Django will be used as it is simple, flexible, reliable and scalable. Analysis of the subsystem in the UML will be done each at a time to ensure that the implementation of the functionality is correct. After thorough analysis, each subsystem will be combined and integrated together to form a complete web application.

4.2.3 Object-oriented design using UML

To provide the graphical representation and documentation of the application design and behaviors, a UML will be used to model the detailed object-oriented design. In Friendstagram, the main functionality is to recommend new potential friends to Nanyang Technological University students who are looking to find like-minded friends. The end-user will be given a questionnaire to fill in their name, hall accommodations, interests and contact information. After that, the information collected will be processed by the Django back-end system. When users click on ‘Find Friends’, the system will send an HTTP request and the user’s information to the server. Once the server gets the request, it will run the recommendation algorithm and generate a list of new potential recommended friends. Apart from that, the Django Authentication System will be implemented to allow end-users to register for an account and login which is needed in order to use the ‘Find Friends’ functionality. After using the ‘Find Friends’ functionality, end-users will also be able to see their history of matched friends.

4.2.3.1 Use Case Diagram

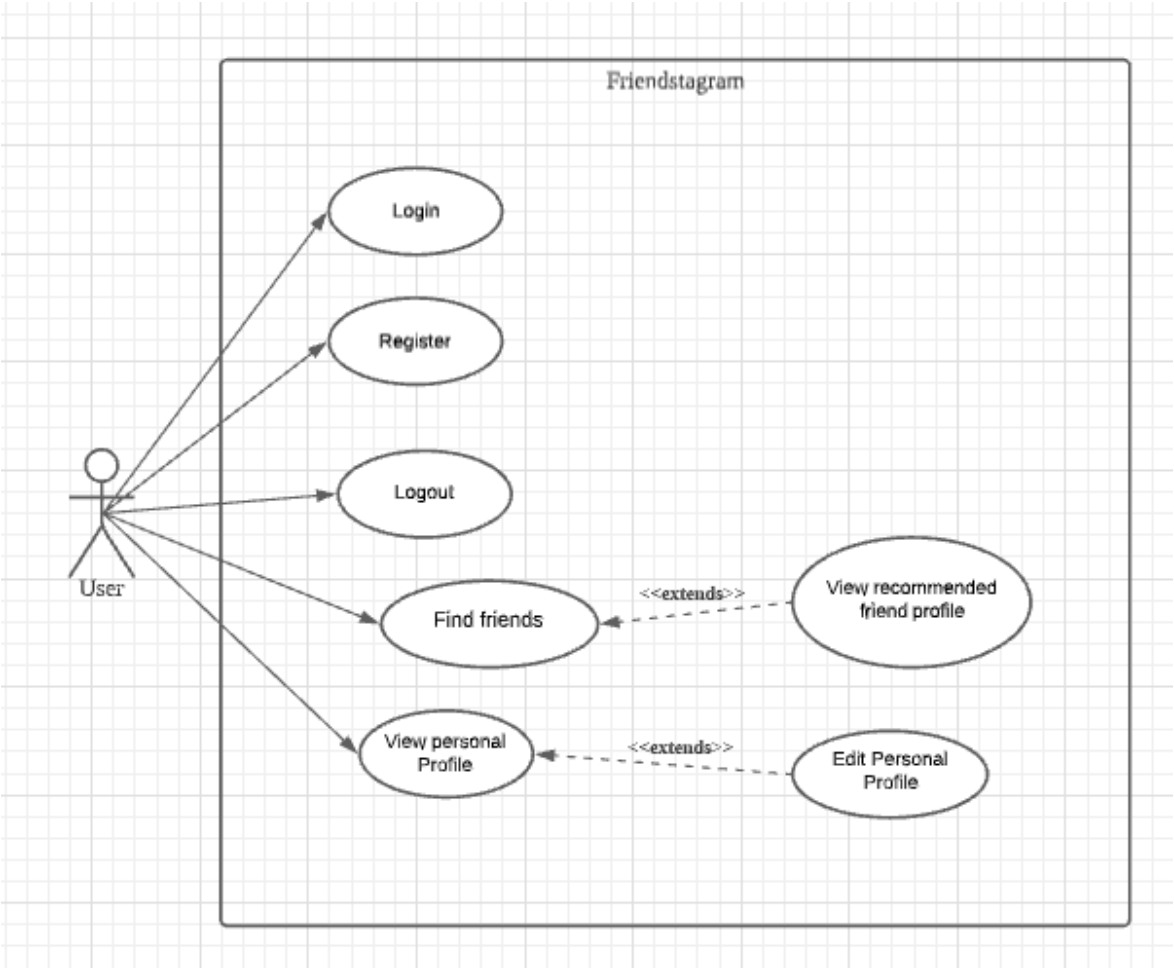

Figure 1. Use Case Diagram

The above Figure 1 is our use case model which defines the system functionalities that we will develop. The use case model will be further elaborated in a separate document, *Use Case Model*.

4.2.4 Prototype

The Friendstagram system will be developed as a web application deployed onto a cloud server and accessible by NTU students on both web and mobile platforms. As our goal is to create a Minimum Viable Product (MVP), only the core functionalities will be developed. We will first draw out a Lo-Fi prototype as seen in Figure 2. A Rapid Prototyping approach will then be employed whereby these core functionalities of the system will be iteratively and incrementally developed in small, separate components before they are integrated together. To further ensure the feasibility of our system, during this rapid prototyping process, the prototype will be deployed onto a test/pre-production server and users' feedback will be sought. Based on users' feedback, modifications to the components will be made to ensure our system addresses users' needs.

Once completed, the prototype will be presented to the implementation team.

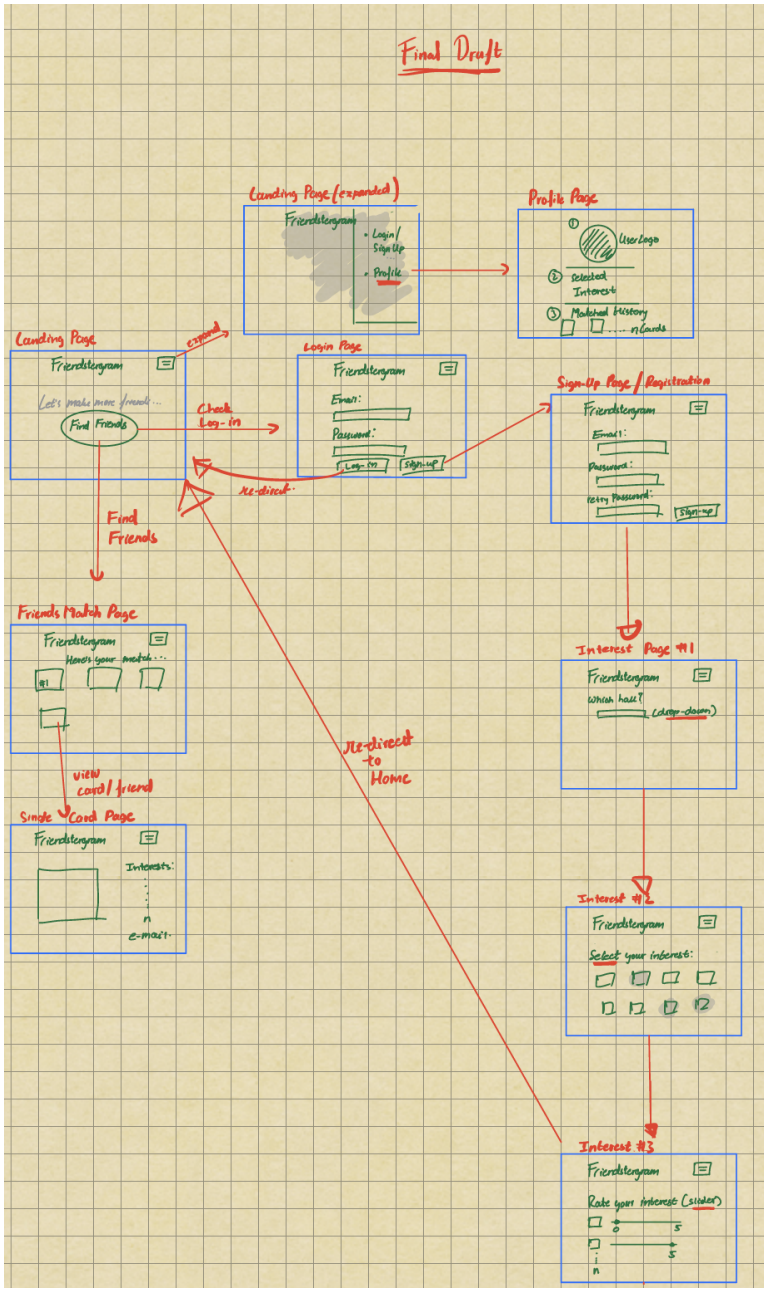

Figure 2: Lo-fi prototype

5 Constraints

5.1 Proprietary Hardware and Software

For the development of Friendstagram, the project team has chosen only free hosting services and

open-source software as seen in Table 1, 2 and 3 respectively. In Table 1, the initial costs for utilizing the online hosting services - Netlify for the frontend and PythonAnywhere for the backend. Both services offer a limited-bandwidth, free plan that is sufficient to run our system. However, if there is a need to scale up as more users use the Friendstagram system, the estimated paid plans are reflected in Table 2. Table 3 shows the libraries and software that we shall utilize in the development of our project which includes React, Redux, Figma, Django, Github, Visual Paradigm, Visual Studio Code, React Testing Library, PyTest and Trello. All of these are either open-source software or free-to-use tools.

The only hardware to be accounted for costs are our laptops used for development which we would be negated as the assumption is that the project team does not pay for/account for personal laptops.

Item	**Supplier**	**Quantity**	**Unit Price($)**	**Total($)**
Front-end server Deployment	Netlify	1	0	0
Back-end server Deployment	PythonAnywhere	1	0	0
			Total:	**0**

Table 1: Online hosting services

Item	**Supplier**	**Quantity**	**Unit Price($)**	**Total($)**
Front-end server Deployment	Netlify	1	19/Mth (4x Bandwidth)	19
Back-end server Deployment	PythonAnywhere	1	5 (100K hits website)	5
			Total:	**24**

Table 2: Online hosting services (paid plans if scale up)

Item	**Supplier**	**Quantity**	**Unit Price($)**	**Total($)**
React Library	Open Source	6	0	0

Redux Library	Open Source	6	0	0
Figma	Figma, Inc.	6	0	0
Django	Open Source	6	0	0
Github	Microsoft	6	0	0
Visual Paradigm	Visual Paradigm International Ltd.	6	0	0
Visual Studio Code	Microsoft	6	0	0
React Testing Library	Open Source	6	0	0
Pytest	Open Source	6	0	0
Trello	Atlassian	6	0	0
			Total:	**0**

Table 2: Open-source softwares

5.2 Project Schedule

The requirements elicitation, design, development and testing of Friendstagram will be conducted over the course of approximately three months, from 24th Jan 2022 to 28th March 2022. The deadline, 28th March 2022 is a hard constraint and cannot be extended. Hence, we will focus on only developing the core functionalities of Friendstagram and if necessary will employ techniques such as *crashing* or *fast-tracking* to ensure we meet the deadline. However, since it is not possible to add new resources/manpower into the project, hence, the project management will from time-to-time switch members' roles around, based on their expertise, to ensure resource usage is maximized and the project can be finished on time.

6 Operational Requirements

6.1 Request Support via Email

Friendstagram users may seek technical or administrative support by sending emails to the Friendstagram administrators. The technical problems may include troubles logging into friendstagram, unspecified errors,

or unexpected server outages or sluggish performance. Administrative support may include reporting users for abusive behaviour or answering questions on how to use Friendstagram.

6.2 Application Services and Technical support

All members in the project team, including non-development members, shall have access to the hosting services and the source code shared via the source code versioning software, GitHub. As members' roles may change from time to time, this is to allow urgent bug fixes to the source code or deployment/hosting services to be made whenever necessary and ensure at least "5 nines" of availability, (99.999% of uptime yearly, or 5.26 minutes of downtime yearly). In order to maintain minimal downtime, network administrators and database support need to be on standby 24/7.

6.3 Administration Features

Friendstagram Administrators must be able to access all registered users' records and manage their accounts such as resetting passwords for users, blocking and deleting accounts. If a user has breached the terms and conditions of Friendstagram or has enacted or displayed behaviour deemed as unbefitting of using services provided by Friendstagram, or has launched attacks on Friendstagram services to disrupt, alter or deny access to services provided by Friendstagram, Friendstagram Administrators must be able to block access for such undesirable users via the backend administrative dashboard.

6.4 System failure and routine backup

System failures must be handled gracefully and services must continue to be provided even when unexpected situations arise, e.g., in the event of an unexpected error causing the backend service to stop executing, the backend service must be able to restart on its own and continue to provide services. Redundancy in our hosting services must also be implemented to ensure that in the event of a regional connection failure, access to Friendstagram services is still available. Additionally, backups of Friendstagram's database must be performed in an automated, regular and consistent manner. The backup disks must also be checked regularly to ensure the backups disks are usable and can be restored. In the event of unexpected data loss, Friendstagram administrators and developers shall be able to restore the data back to the pre-incident state.

6.5 Audit Trail

An audit trail must be implemented in the Friendstagram system to ensure **non-repudiation** and

accountability of users' actions. This includes logging every user action and logs must be stored for a minimum of 2 years. User actions may include creating new accounts, logging in, logging out, making new friend-finding requests and so on. The data shall include the date, time, IP addresses and MAC addresses. The audit trail is important in ensuring users may not deny having performed an action and detecting suspicious behaviour. It also enhances our efforts in preventing cyber security incidents from taking place.

7 Functional Requirements

This web application is a platform that helps users connect with new people with similar interests and hobbies. Do achieve this, the recommendation web application should have these main functional requirements:

7.1 Register

The web application must allow the user to register for an account.

7.2 Login

The web application must allow the user to log in to have access to all features.

7.3 Logout

The web application must allow the user to log out from the platform.

7.4 Recommend Friends

The web application must recommend the top k most similar profiles based on the user personal information like hall residences, interests and contact information, etc.

7.5 View Recommended Profiles

The web application must allow the user to view recommended profiles listing their interests, contact information, etc.

7.6 View Personal Profile

The web application must allow the user to view their own profile that reflects the user's interests, hall residences information etc.

7.7 View Friends Matched History

The web application must show a history of matched friends in their personal profile page.

7.8 Edit Personal Profile

The web application must allow the user to edit/modify/change their interests, etc in their profile.

8 Input Requirements

8.1 Email Address and Password

The user must input their email address and password to register an account.

8.2 About Me, Interests and Hall

The user must input their ‘About Me’, interests and hall accommodation information before they can get recommendations on new potential friends.

9 Process Requirements

The following are among the inherent requirements that the web application must be able to handle:

9.1 Backend Service Transactions via HTTP

The frontend client application must be able to send and receive HTTP requests to the backend service.

9.2 Data Integrity

The system must utilize SSL certificates to encrypt communications between the frontend and backend systems, i.e. data must not be modified by a malicious attacker performing a man-in-the-middle attack.

9.3 Data Validation

Data validation must be performed on both the client and server side to ensure invalid or malicious data input cannot be injected and/or forcibly stored into the Friendstagram system. Invalid data inputs must not be cached and buffers must be cleared properly. Proper error handling must also be performed for all functionalities in the Friendstagram system.

9.4 Performance

The Friendstagram system must be able to handle concurrent requests from multiple users without crashing. All errors must be handled gracefully without crashing the system and all errors must be displayed back to the user without disclosing the internal architecture of the Friendstagram system, e.g., backend server error codes and messages must not be displayed in its raw form back to users to prevent unintentional leakage of sensitive data of Friendstagram system. In the event of an unexpected crash, the system must be able to restart on its own and continue to provide services.

10 Output Requirements

10.1 List of Recommended Friends

The system must display a list of recommended friends based on their common interests and profile characteristics.

10.2 Personal Profile and History of Matched Friends

The system must display the profile of the user and the history of his/her matched friends.

10.3 Potential Friends Profile

The system must display the profile of the potential friend chosen by the user which includes their interests, hall residence and contact information.

11 Hardware Requirements

11.1 Client Computers and Mobile Devices

Any computer and mobile device that has a compatible browser installed.

11.2 Network

H/H+/3G/4G/5G network connection or 2.4GHz/5GHz WIFI connection with Internet access is required in order to be accessed.

12 Software Requirements

12.1 Client Operating Systems

- Android, Windows, macOS, Linux, iOS to load the web application

12.2 Network system

The system built must be compatible with Local Area Network (LAN) and Wireless Local Area Network (WLAN) standards defined in IEEE 802.11 for network communications, e.g., Django backend and the React frontend client. The system must support the following protocols:

- HTTP
- HTTPS
- TCP/IP

12.3 Licenses

In order to use software development tools from third-party vendors such as Figma, Visual Paradigm, Pycharm, Visual Studio Code, valid licenses are needed. Apart from that, using third-party packages, web servers and database software require valid licenses too.

13 Deployment Requirements

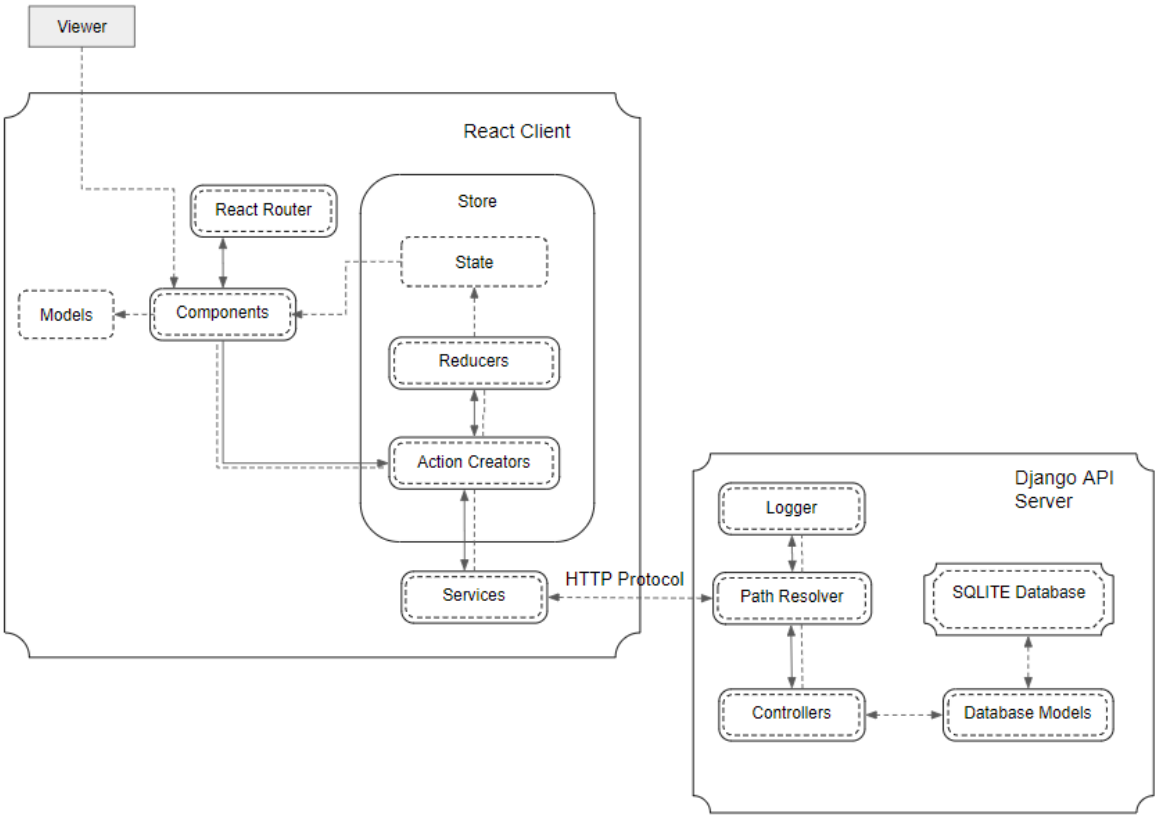

Figure 3. System Architecture of Friendstagram