Inspiration

Friendstagram

[Project Plan]
Version 1.0

By: Royce Ang Jia Jie, Manav Arora, Jovan Huang Tian Chun, Clarence Hong Shi Man, Zhu Weiji, Tan Hui Zhan

Revision History

Revision Number	Date	Primary Author(s)	Comments
0.1	Feb 21st, 2022	Manav Arora, Jovan Huang Tian Chun, Clarence Hong Shi Man, Tan Hui Zhan	First version
0.2	Mar 13th, 2022	Manav Arora, Jovan Huang Tian Chun, Clarence Hong Shi Man, Tan Hui Zhan, Zhu Weiji, Royce Ang Jia Jie	Completed Project Plan
1.0	Mar 13th, 2022	Manav Arora	Reviewed and approved Project Plan

Table of Contents

1 Introduction

1.1 Project Overview

Friendstagram is a responsive web-application that acts as a friend recommendation system for Nanyang Technological University students. Given that students are usually on their mobile phones, a web-application that is suitable for mobile will be built. Based on users' personal interests and hall accommodation information, Friendstagram will generate a list of new potential friends for users to reach out to.

1.2 Project Description and Scope

Friendstagram is an idea that arose during a discussion among Nanyang Technological University students whose daily lives have been disrupted by COVID-19 pandemic. Due to COVID-19 pandemic, many social restrictions have been put in place to prevent the spread of the virus amongst student cohorts. The inability to interact with others face-to-face has caused students to achieve a less fulfilling university life. Hence, Team Inspiration decided to create Friendstagram, a web application to help students cope with this situation. Team Inspiration will be in charge of both the front-end and back-end development of this web application.

The basis of Friendstagram is a friend recommendation system in the form of a responsive web application that takes a number of possible constraints into account. These can be such limits as:

- The user intends to find new potential friends who stay in the same hall;
- The user intends to find new potential friends who likes playing badminton;
- The user intends to find new potential friends who loves to study;
- It is important that users can choose multiple interests;
- The users might have greater preference for one interest over another;
- The user must be able to change their interests from time to time.

The above are just some possible examples of the constraints. There will be more discussion on the final possible constraints in a later document.

Given these preferences, Friendstagram will allow users to key in their personal interests and hall accommodation information before generating a list of potential friends. However, as this project involves only Nanyang Technological University students and is a Minimum Viable Product (MVP), functionalities such as real-time texting are beyond the scope. These functionalities can be considered after the User Acceptance Testing of the MVP is completed where we have users' feedback. Team Inspiration will be providing the deliverables as stated in this document which includes all the source code, lo-fi prototype, use case model, a test data set and database schema.

Friendstagram is responsible for generating a list of potential friends for users after they have logged in and keyed in their personal interests and hall accommodation information. Friendstagram's responsibilities are complete once the list of potential friends and their personal information like interests and hall accommodation are generated for the users. This allows the users to reach out to these new potential friends.

The system of Friendstagram will include all the necessary user interfaces to quickly and intuitively determine the user's requirements and preferences.

Also, since Friendstagram is only targeted at Nanyang Technological University students, users will have to use their school email to sign up for an account on our platform.

2 Project Organization

2.1 Team Structure

The following is the list of executive roles:

- Project Manager: Manav Arora
- QA Manager: Jovan Huang Tian Chun
- QA Engineer - Tan Hui Zhan
- Lead Developer: Royce Ang Jia Jie
- Front-end Developer - Royce Ang Jia Jie
- Back-end Developer - Zhu Weiji
- Release Engineer/Manager - Clarence Hong Shi Man

2.2 Roles and Responsibilities

Project Manager: Manav Arora

- Oversees project progress
- Approves and executes project plan
- Assigns tasks and reports status of project to team members
- Manages and motivates team members
- Represents the team to the outside world
- Ensures overall delivery of product is completed smoothly

QA Manager: Jovan Huang Tian Chun

- Ensure that overall product and process quality are of high standards
- Implement QA processes

QA Engineer: Tan Hui Zhan

- Devise test plans
- Conduct comprehensive tests

Lead Developer: Royce Ang Jia Jie

- Overall technical lead
- Responsible for technical aspects of product release
- Ensure that the final system design is of high standards

Front-end Developer: Royce Ang Jia Jie

- Conduct front-end programming for web application
- Participates in the entire SDLC
- Generates work products including documentation, source code, unit and integration tests

Back-end Developer: Zhu Weiji

- Server, application and database programming
- Participate in the entire SDLC
- Generates work products including documentation, source code, unit and integration tests

Release Engineer/Manager: Clarence Hong Shi Man

- Create baselines and build and integrate changes for delivery
- Manage releases of the product prototype

2.3 Team Communication

Team Inspiration communication channels include the following:

- Weekly meetings are held on the online Zoom platform or in person when necessary, either on Monday or Wednesday.
- Important information and updates are announced via Discord and Whatsapp.
- Zoom, Discord and Whatsapp messaging are used to facilitate discussions and clarify doubts.
- Subgroups are formed in order to facilitate stronger focus and cooperation in completing the subtasks.
- Github is used to share codes and facilitate clear communication on the status of every issue.
- Google Drive is used to help facilitate real-time files sharing and documentation editing, allowing real-time communication of ideas and collaboration between multiple team members.
- Backlogs are used to communicate outstanding tasks and task's status.

3 Process Definition

3.1 Lifecycle Model

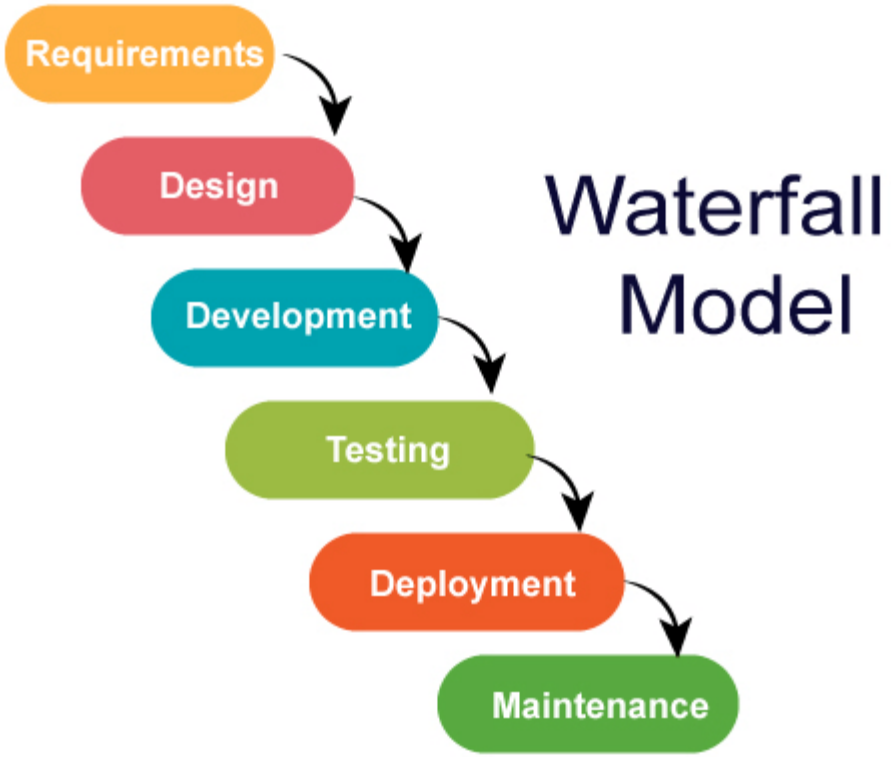


Fig 1. Waterfall SDLC Model Illustration

Team Inspiration intends to use the Waterfall Software Development Life Cycle (SDLC) Model from the gathering of requirements to the maintenance of the project, Friendstagram. In this technique, the development cycle is broken down into different phases wherein each phase is connected to the next sequentially, i.e. output of one serves as input to another.

This methodology has allowed us to speedily develop a product that is ready for release in a short amount of time. This method allows us to adhere to deadlines and keep the project on track. It helps us set out clear and concise requirements from an early stage which facilitate the development process. Overall, this method allowed us to better manage our time and consolidate a streamlined effort to complete this project.

4 Schedule

4.1 Activity Dependencies and Schedule

Milestone Description	Category	Progress	Start Date	End Date
Project Planning(Proposal)			24/1/22	6/2/22
Assignment of Team Roles	On Track	100%	24/1/22	25/1/22
Identify Problem	On Track	100%	25/1/22	26/1/22
Elicit Requirements	On Track	100%	26/1/22	29/1/22
Identify Technologies to be used	On Track	100%	26/1/22	27/1/22
Use Case Modeling	On Track	100%	27/1/22	30/1/22
Design System Architecture	On Track	100%	30/1/22	5/2/22
Design User Interface	On Track	100%	27/1/22	6/2/22
Backend Development			7/2/22	28/2/22
Design Database Structure and Schema	On Track	100%	7/2/22	9/2/22
Deploy Server to Cloud	On Track	100%	9/2/22	10/2/22
Develop API Endpoints			12/2/22	28/2/22
Authentication Endpoints	On Track	100%	12/2/22	17/2/22
Friends Management Endpoints	On Track	100%	17/2/22	23/2/22
Groups Management Endpoints	On Track	100%	23/2/22	28/2/22
Frontend Development			7/2/22	28/2/22
Authentication Pages	On Track	100%	7/2/22	13/2/22
Friends Management Pages	On Track	100%	15/2/22	21/2/22
Groups Management Pages	On Track	100%	21/2/22	27/2/22
System Integration			27/2/22	17/3/22
Integration of Authentication Pages	On Track	100%	27/2/22	5/3/22
Integration of Friends Mgmt Pages	On Track	100%	5/3/22	11/3/22
Integration of Groups Mgmt Pages	On Track	100%	1/3/22	16/3/22
Quality Planning and Control			6/2/22	9/3/22
Implement testing for Authentication	On Track	100%	17/2/22	20/2/22
Implement testing for Friends Mgmt	On Track	100%	23/2/22	26/2/22
Implement testing for Groups Mgmt	On Track	100%	28/2/22	3/3/22

Implement Integration Testing	On Track	100%	5/3/22	8/3/22
Other Documentation			7/2/22	28/3/22
System Requirement Specification	On Track	100%	7/2/22	20/2/22
Project plan	On Track	100%	7/2/22	20/2/22
Quality plan	On Track	100%	21/2/22	6/3/22
Risk management plan	On Track	100%	21/2/22	6/3/22
Design for software maintainability report	On Track	100%	7/3/22	20/3/22
Configuration Management Plan	On Track	100%	7/3/22	20/3/22
Change Management Plan	On Track	100%	7/3/22	20/3/22
Release Plan	On Track	100%	7/3/22	20/3/22
Test Plan	On Track	100%	21/3/22	27/3/22
Test Cases	On Track	100%	21/3/22	27/3/22
Requirements Test Coverage	On Track	100%	21/3/22	27/3/22

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**Friendstagram**	59 days	Mon 24/1/22	Mon 28/3/22	
2		**Project Planning (Proposal Writing)**	**13 days**	**Mon 24/1/22**	**Sun 6/2/22**	
3		Assignment of Team Roles	1 day	Mon 24/1/22	Tue 25/1/22	
4		Identify Problem	1 day	Tue 25/1/22	Wed 26/1/22	3
5		Elicit Requirements	3 days	Wed 26/1/22	Sat 29/1/22	4
6		Identify Technologies to use	1 day	Wed 26/1/22	Thu 27/1/22	4
7		Use Case Modelling	3 days	Thu 27/1/22	Sun 30/1/22	6
8		Design System Architecture	5 days	Sun 30/1/22	Sat 5/2/22	7
9		Design User Interface	10 days	Thu 27/1/22	Sun 6/2/22	6
10						
11		**Backend Development**	**20 days**	**Mon 7/2/22**	**Mon 28/2/22**	
12		Design Database Structure and Sche	2 days	Mon 7/2/22	Wed 9/2/22	
13		Deploy server to cloud	1 day	Wed 9/2/22	Thu 10/2/22	12
14		**Develop API Endpoints**	**15 days**	**Sat 12/2/22**	**Mon 28/2/22**	**13**
15		Authentication Endpoints	5 days	Sat 12/2/22	Thu 17/2/22	
16		Friends Recommendation Endpoi	5 days	Thu 17/2/22	Wed 23/2/22	15
17		Profile Endpoints	5 days	Wed 23/2/22	Mon 28/2/22	16
18		**Frontend Development**	**20 days**	**Mon 7/2/22**	**Mon 28/2/22**	
19		Authentication Pages	6 days	Mon 7/2/22	Sun 13/2/22	
20		Friends Recommendation Pages	6 days	Tue 15/2/22	Mon 21/2/22	19
21		Profile Pages	6 days	Mon 21/2/22	Sun 27/2/22	20
22						
23		**System Integration**	**16 days**	**Sun 27/2/22**	**Thu 17/3/22**	**14,18**
24		Integration of Authentication Pages	5 days	Sun 27/2/22	Sat 5/3/22	
25		Integration of Friends Recom. Page	5 days	Sat 5/3/22	Fri 11/3/22	24
26		Integration Profile Pages	5 days	Fri 11/3/22	Wed 16/3/22	25
27						
28		**Quality Planning and Control**	**28 days**	**Sun 6/2/22**	**Wed 9/3/22**	**2**
29		Implement testing for Authenticatic	3 days	Thu 17/2/22	Sun 20/2/22	15
30		Implement testing for Friends	3 days	Wed 23/2/22	Sat 26/2/22	16,20
31		Implement testing for Profile	3 days	Mon 28/2/22	Thu 3/3/22	17,21
32		Implement Integration testing	3 days	Sat 5/3/22	Tue 8/3/22	24
33						
34		**Other Documentation**	**46 days**	**Mon 7/2/22**	**Mon 28/3/22**	
35		System Requirements Specification	13 days	Mon 7/2/22	Sun 20/2/22	2
36		Project Plan	13 days	Mon 7/2/22	Sun 20/2/22	2
37		Quality Plan	12 days	Mon 21/2/22	Sun 6/3/22	35,36
38		Risk Management Plan	12 days	Mon 21/2/22	Sun 6/3/22	35,36
39		Design for Software Maintainability Report	12 days	Mon 7/3/22	Sun 20/3/22	37,38
40		Configuration Management Plan	12 days	Mon 7/3/22	Sun 20/3/22	37,38
41		Change Management Plan	12 days	Mon 7/3/22	Sun 20/3/22	37,38
42		Release Plan	12 days	Mon 7/3/22	Sun 20/3/22	37,38
43		Test Plan	6 days	Mon 21/3/22	Sun 27/3/22	39,40,41,42
44		Test Cases	6 days	Mon 21/3/22	Sun 27/3/22	39,40,41,42
45		Requirements Test Coverage	6 days	Mon 21/3/22	Sun 27/3/22	39,40,41,42
46		CMMI Level 2 Definition	6 days	Mon 21/3/22	Sun 27/3/22	39,40,41,42

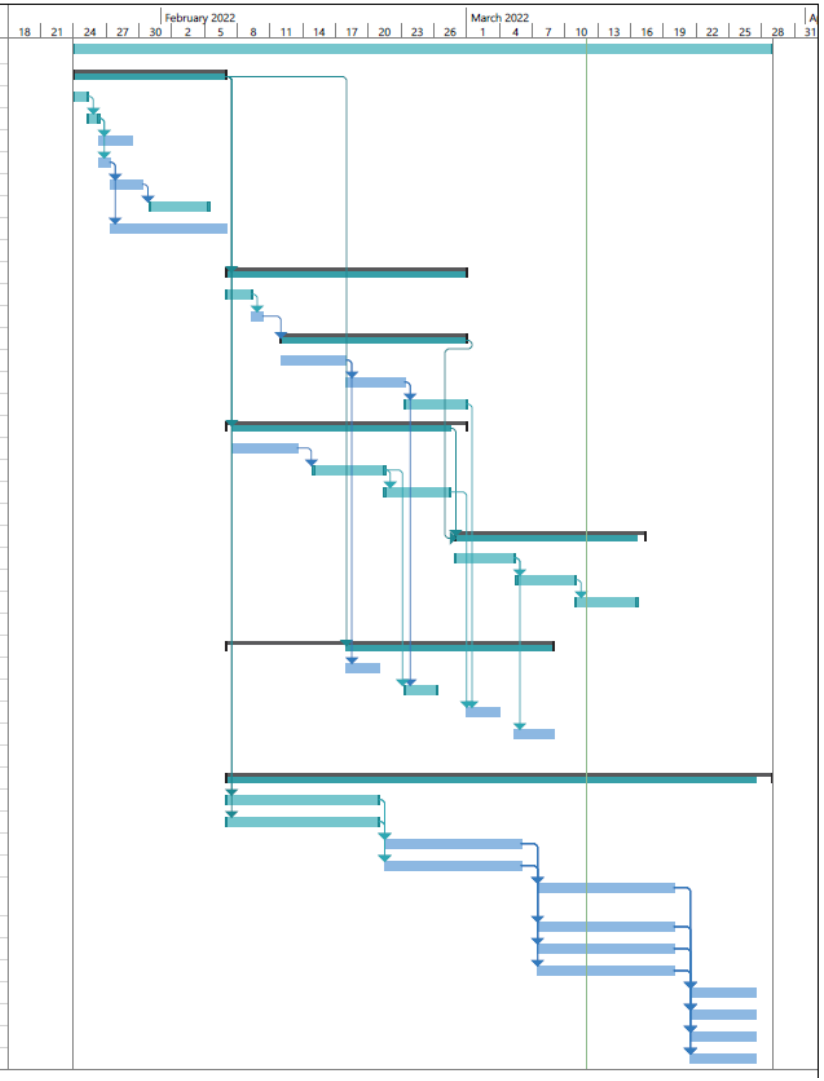


Fig 2. Gantt Chart of Friendstagram Project

4.2 Work Breakdown Structure

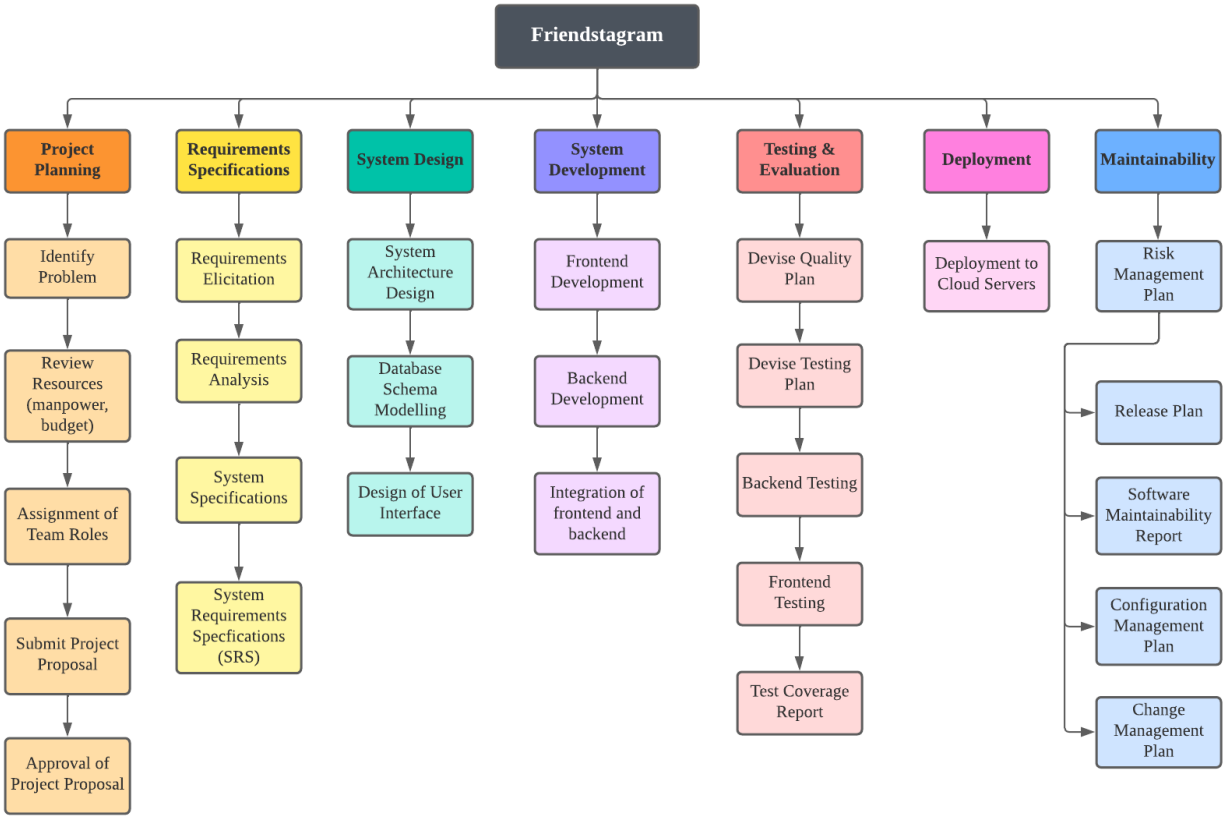


Fig 3. Work Breakdown Structure of Friendstagram

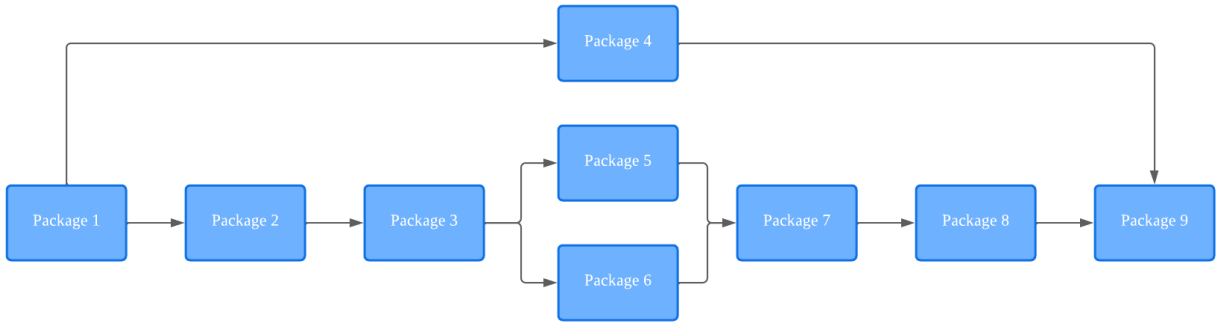
4.3 Work Packages

The work for the project, Friendstagram will be broken into different phases and milestones of the software development life cycle. They are as follows:

1. Project Planning
2. Requirement Specification
3. System Design
4. Documentation
5. Front-end development
6. Back-end development
7. System Integration
8. System Testing
9. Project Handover (submission) to Client

4.4 Activity Dependencies

Activity Network Diagram + Work Package & Activity Table



No.	Description	Start Date	Duration	End Date	Dependencies	Comments
1	Project Planning	24/1/22	13 days	6/2/22	-	
2	Requirement Specifications	7/2/22	13 days	20/2/22	1	
3	System Design	27/1/22	11 days	6/2/22	2	
4	Documentation	6/2/22	76 days	11/4/22	1	Occurs concurrently with others
5	Frontend Development	7/2/22	20days	28/2/22	3	Occurs concurrently with 6
6	Backend Development	7/2/22	20days	28/2/22	3	Occurs concurrently with 5
7	System Testing	17/2/22	29 days	17/3/22	5, 6	
8	System Integration	27/2/22	18 days	17/3/22	7	
9	Project Handover (submission) to Client	27/3/22	1 day	27/3/22	4,8	

testing started last lab so it should be march

4.5 Work Package Details

Project	Friendstagram
Work Package	1 – Project Planning (1 of 9)
Assigned To	All
Effort	13 Days
Start Date	24/01/22
End Date	06/02/22
Purpose	Choose a project topic, and determine overall direction of the project which will be consistently updated in future work packages.
Input(s)	-
Activities	This work package involves distributing team roles and identifying the problem and technologies to be used.
Output(s)	A written document of the Project Proposal and Use Case Model

Project	Friendstagram
Work Package	2 - Requirement Specification
Assigned To	All
Effort	13 Days
Start Date	07/02/22
End Date	20/02/22
Purpose	List down the features needed and distribute work
Input(s)	Package 01

Activities	This work package involves doing feasibility study, requirements elicitation, constraints evaluation and project schedule planning. It also involves providing the operational requirements, functional requirements, input requirements, process requirements, output requirements, hardware requirements, software requirements and deployment requirements.
Output(s)	A written document of System Requirement Specification

Project	Friendstagram
Work Package	3 – System Design
Assigned To	All
Effort	11 days
Start Date	27/01/22
End Date	06/02/22
Purpose	To design a backend architecture to support front-end development and discuss on overall system design
Input(s)	Package 02
Activities	Planning structure and schema of database, setting up the server
Output(s)	Database schema, high-level design and architectural specification

Project	Friendstagram
Work Package	4 - Documentation
Assigned To	All
Effort	76 Days
Start Date	06/02/22
End Date	11/04/22
Purpose	Document everything, from Project Proposal to UI/UX Design to Unit Test

Input(s)	Package 01
Activities	Writing documentation for every stage of developing software.
Output(s)	Project Proposal, Use Case Description, Quality Plan, System Requirement Specification, Project Plan, Risk Management Plan, Test Plan, Test Cases & Requirement Test Coverage Report. Front-end mockup and wireframing, Recommendation algorithm, Back-end Design

Project	Friendstagram
Work Package	5 - Frontend Development
Assigned To	Royce, Hui Zhan
Effort	20 Days
Start Date	07/02/22
End Date	28/02/22
Purpose	Design the user interface between the system and the users, to make it easy to use, and implement the front-end of the application.
Input(s)	Package 03
Activities	Front-end mockup and wireframing, Discuss flowchart for questionnaire, Implement login and signup pages, main application pages, and mobile responsiveness. Implement modules according to the design specifications noted in the specification document
Output(s)	Front-end mockup and wireframing, Source code and logical files, Working front-end design ready for integration

Project	Friendstagram
Work Package	6 - Backend development
Assigned To	Weiji
Effort	20 Days

Start Date	07/02/22
End Date	28/02/22
Purpose	Implement back-end structure
Input(s)	Package 03
Activities	Analyze data flow relationships, entity relationships, implement database and API calls
Output(s)	Working back-end server ready for integration

Project	Friendstagram
Work Package	7 - System Testing
Assigned To	All
Effort	29 Days
Start Date	17/2/22
End Date	17/3/22
Purpose	Implement unit testing
Input(s)	Package 07
Activities	Simulate how a user might interact with the system. Testers will look out for system performance and integrity and ensure the program meets requirement specifications.
Output(s)	Test report

Project	Friendstagram
Work Package	8 - System Integration
Assigned To	Royce, Hui Zhan, Weiji

Effort	18 Days
Start Date	27/2/22
End Date	17/3/22
Purpose	Integrate the front-end and back-end
Input(s)	Package 05, 06
Activities	Integration of login page and registration page, profile page and recommendation page.
Output(s)	Working Web-Application prototype

Project	Friendstagram
Work Package	9 - Project Handover
Assigned To	All
Effort	1 Day
Start Date	27/03/22
End Date	27/03/22
Purpose	Finish all documentations and handover to client
Input(s)	All packages
Activities	Handover all documentations and codes to client Post-project Review - reflect on what was done well and mistakes made (how to avoid them in future projects)
Output(s)	All

5 Project Estimates

5.1 Code Size Estimation using Function Points

Unadjusted Function Points

Friendstagram have the following functionalities:

Users:

- Register
- Login
- Logout
- Find Friends
- View recommend friend profile
- View personal profile
- Edit personal profile

Unadjusted function points are measured based on 5 primary component elements of the following functions: Inputs, Outputs, Inquires, Logical Files, and Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity. The detailed evaluation of the complexity is as follows:

Rating Inputs:

- Register for an account (Email, Password)
- Gathering user's information after registration (Hall Accomodation, Interests, Interests Ratings, About Me)

Rating Outputs:

- Displaying a list of recommended friends that matches user's profile (Images, Interests)
- Displaying the profile of a recommended friend (Contains Email, About Me, Hall Accommodation Details, Interests)
- Displaying user's personal profile information (Email, Hall Accommodation Details, Interests)
- Displaying history of matched friends

Rating Inquires:

- Selecting user account when user is logging in
- Selecting a list of recommended friends that matches user's profile
- Selecting a recommended friend profile

Rating Logical Files:

- User Account (Email, Password, Hall Accommodation Details, Interests, About Me)
- User Personal Information (Email, Hall Accommodation Details, Interests, About Me)

Rating Interfaces:

- One External Generator used (Images)

Summary of the above analysis:

Element	**Complexity**	**Detail**
Inputs	Low	Register for an account
	Low	Gathering user's information after registration
Outputs	High	Displaying a list of recommended friends
	Low	Displaying the profile of a recommended friend
	Low	Displaying user's personal profile information
	Low	Displaying history of matched friends
Inquiries	Low	Selecting user account when user is logging in
	High	Selecting a list of recommended friends that matches user's profile
	Low	Selecting a recommended friend profile
Logical Files	Low	User Account
	High	User Personal Information
Interfaces	Low	External Image Generator API

Calculation of Unadjusted Function Points:

Characteristic	Low Complexity	Medium Complexity	High Complexity
# Inputs	2 x 3 = 6	0 x 4 = 0	0 x 6 = 0
# Outputs	3 x 4 = 12	0 x 5 = 0	1 x 7 = 7
Inquiries	2 x 3 = 6	0 x 4 = 0	1 x 6 = 6
Logical Files	1 x 7 = 7	0 x 10 = 0	1 x 15 = 15
Interfaces	1 x 5 = 5	0 x 7 = 0	0 x 10 = 0
Unadjusted FP	36	0	28
Total = L + M + H	64		

Adjusted Function Points

Influence Factors	Score (0-5)	Detail
Data communications	5	The web application has both front-end and back-end and it supports 24/7 communication with Django Server, which allows it to function properly.
Distributed functions	4	Distributed processing and data transfer are unidirectional between the front-end side and the backend servers.
Performance	3	Response time / throughput is important and critical to the part of the application where recommendations of potential friends are generated. There is no need to have a special design for CPU utilization.
Heavily used	1	Users will use the application only when they want to find new friends. Although the load on the system is likely to be light, we took timing into consideration where the application needs to generate the list of recommended friends in 1-3 seconds.
Transaction rate	2	Throughout the day, transactions are anticipated to be relatively constant.

On-line data entry	4	On-line data entry is extremely important for the web application. We need users to input their personal information which will be used to generate a list of new recommended friends.
End-user efficiency	4	We have a few efficient designs to make the web application intuitive and easy for end users to use and navigate.
On-line update	2	We support online updates of major internal logical files although it is not very frequent. It only occurs when new users sign up for an account or update their personal information.
Complex processing	5	Friendstagram's recommendation system has a complex algorithm behind it.
Reusability	2	Our application is currently a Minimal Viable Product (MVP), and is at a much smaller scale. Therefore, there is less emphasis on reusability.
Installation ease	0	Our prototype is a web-app, and hence need not be installed.
Operational ease	3	The web application is designed for unattended operation. Unattended operation means no human intervention is required to operate the system other than to start up or shut down the application.
Multiple sites	1	Our web application only has one site.
Facilitate change	3	Our web application supports changes when users update their personal information. This will update the internal logical files and will not change any of the functionalities or the algorithm behind our friend recommendation system.
Total	39	

Scoring (0 – 5)
0 = No influence

1 = Insignificant influence
2 = Moderate influence
3 = Average influence
4 = Significant influence
5 = Strong influence

Influence multiplier = 39 x 0.01 + 0.65 = 1.04

Adjusted Function Point Total: 64 x 1.04 = 66.56

Lines of Code

For the frontend codes, we used HTML and JavaScript / ES6.

The average number of source lines per FP for HTML and Javascript / ES6 is 34 and 47 respectively.

For the backend codes, we used Python and SQL.

The average number of source lines per FP for Python and SQL is 24 and 21 respectively.

Overall, the average number of source lines per FP for our project is 31.5.

Therefore, we have: **Lines of Code** = 66.56 FP X 31.5 LOC/FP = 2097 LOC.

5.2 Effort, Duration & Team Size Estimation

In order to estimate the effort and duration needed to complete this project, we use function points to calculate Effort, Duration and Team size and finally the schedule. We make sure that these estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates.

Based on our team's history projects, we have an average of 200 LOC/PM.

Hence, the estimated effort, duration and team size is as follows:

Effort = (2097 LOC) / (200 LOC/PM) = 10.49 PM

Duration = 3.0 x 10.49^(0.33) = 2.17 months

Team Size = 5.99 PM / 2.17 months = 2.76 people

5.3 Cost Estimates

Hardware:

Developer workstations:

Macbook Pro	
8-Core CPU	
14-Core GPU	Total $3000.00
16GB Unified Memory	
Apple M1 Pro Chip	
512 GB SSD Drive	

Software:

Development Tools:

PythonAnywhere Web Server	$0.00
React library	$0.00
Django library	$0.00
Python	$0.00
JavaScript	$0.00
GitHub	$0.00
Netlify	$0.00
PythonAnywhere	$0.00

Documentation Tools:

Visual Paradigm	$0.00
Microsoft Visual Studio Code	$0.00
Microsoft Office	$0.00

6 Product Checklist

Items listed below will be delivered on the stated deadlines.

No.	Project Deliverable	Estimated Deadline
1	Project proposal, Use case model	February 7, 2022
2	System Requirement Specification (SRS)	February 21, 2022
3	Quality Plan	February 21, 2022
4	Project Plan	March 14, 2022
5	Risk Management	March 14, 2022
6	Design Report on software maintainability	March 28, 2022
7	Configuration Management Plan	March 28, 2022
8	Change Management Plan	March 28, 2022
9	Release Plan	March 28, 2022
10	Test Plan	April 11, 2022
11	Test Cases and Requirements Test Coverage report	April 11, 2022
12	CMMI level 2 definition	April 11, 2022

7 Best Practice Checklist

No.	Practices	✔
1	Documentation must be in a standardized format.	
2	Weekly meetings should be held in accordance with the Waterfall SDLC Model. Each member should be made aware of any changes.	
3	Changes in code must be checked by a fellow developer and must pass the specific test case, before it can be merged.	
4	Bugs and errors should be highlighted to all team members.	
5	Coding Style must be consistent and follow the preset guidelines.	
6	Project Scheduling should be done thoroughly and followed so that the project can be completed in a timely manner while meeting requirements.	
7	Software testing should be high priority and very thorough to ensure product quality and standards.	

8 Risk Management

The following risks have been identified for the Friendstagram project:

Team member unavailable before project completion
Probability: Low
Impact: High
Impacts: There will be a lack of manpower required to complete the project on time. Remaining team members will have increased workload to meet the project deadline, leading to increased stress levels and potentially conflicts.
Risk Reduction: Team members are given benefits for their hard work.

Requirements miscommunication
Probability: Medium
Impact: High
Impacts: This can lead to increased workload due to the development of features that are not required or missing out required features of the software. This will result in missing the deadline for the project due to the increased workload.
Risk Reduction: Have meetings at least once a week to reduce the probability of miscommunication. Team members must inform the team of any new updates and development knowledge.

Internal conflict between teammates
Probability: Low
Impact: Medium
Impacts: Work productivity will drop as teammates are unwilling or avoid communicating with one another. Internal conflicts will lead to friction that reduces the overall productivity of the team.
Risk Reduction: Always communicate any issues with one another before it leads to a conflict. All conflicts must be resolved as soon as possible.

Deadline delayed due to other assignments
Probability: Medium
Impact: High
Impacts: Assignments and tests will reduce the amount of time each team member can spend on the development of Friendstagram. If the reduced time is not taken into account during the project scheduling, deadlines will be delayed as there will not be enough time to finish the project on time.
Risk Reduction: Each member must inform the team of their assignment and test once they have the information. Adjust the project schedule based on the availability of each member and do more work on less busy weeks.

Lack of physical team meetings
Probability: High
Impact: Low
Impacts: Unable to discuss project progress physically, leading to lack of awareness on each team member's progress on their assigned workload.
Risk Reduction: Have team meetings on online platforms such as discord and zoom. Ensure

constant communication and updates through messaging platforms such as WhatsApp and Telegram.

9 Quality Assurance

To maintain a certain high quality standard for our project, the following quality assurance scheme has been devised. The following check will be conducted:

- **Unit Testing**

  Every component will be tested to make sure that they are performing as expected.

- **System Testing**

  The functionality will be checked to make sure that the application performs as intended.

- **In-Place Testing**

  The application will be tested as a whole.

- **Acceptance Testing**

  This will be done to ensure the application meets the requirements specification.

The specific procedures and details of the above testing methodologies will be elaborated on in the Test Plan.

10 Monitoring & Control

A multitude of features and processes are put in place to be able to successfully monitor the progress of this software project. Some of the essential ones are mentioned below:

Measurement of resource consumption:
This project involves a plethora of human resources in order to achieve the goal set out by the team. This involves efforts put into the code in terms of frontend, backend, algorithm, etc. Keeping keen notice of the resources used for each milestone allows for a smoother and successful implementation.

Identification of major project risks:
Any potential risks to the project must be identified in advance in order to come up with mitigation strategies. Detailed information on this topic is provided in the Risk Management section of this document.

Regular reviews of project progress:
The project team meets every week in order to discuss and review the status of the project while covering topics regarding code development, code testing, documentation overview, backlog etc. Physical meetings are conducted at the lab which allows for a face to face discussion on the issues mentioned above.

Timeline Planning and task decomposition:
This document outlines the estimated timeline for this project. A rough timeline can be estimated whilst breaking down the various tasks into sub-tasks, allowing for periodical reviews of the project progress. Project subcomponents and timeline estimates are included in the Schedule section of this document.