

Project Plan

for

Wanderers

Hazim Khoiruddin
Isaac Chun Jun Heng
J'sen Ong Jia Xuan
Kim Seo Jin
Leong Mininn Miko
Raghav Rajendran Nair
Yu Zi Hao Albert

Team Wanderers
College of Computing and Data Science, Nanyang Technological University

Submitted to—
Goh Tong Hai
Li Shenggui
College of Computing and Data Science, Nanyang Technological University

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	March 10 th 2025	Raghav Rajendran Nair	First Version
1.1	March 11 th 2025	Isaac Chun Jun Heng	Refined entire document and added project estimates using FP
1.2	March 12 th 2025		Added more realistic function point calculation and refined language of document and format.

Table of Contents

Revision History	2
Table of Contents.....	3
1. Introduction	5
1.1 Project Overview	5
1.2 Project Description and Scope	5
2. Project Organization	6
2.1 Team Structure	6
2.2 Roles and Responsibilities	6
2.3 Team Communication	7
3. Process Definition	8
3.1 Life Cycle Model	8
4. Schedule	9
4.1 Activity Dependencies and Schedule.....	9
4.2 Work Breakdown Structure	11
4.3 Work Packages	12
4.4 Activity Dependencies	12
4.5 Work Package Details.....	13
5 Project Estimates	16
5.1 Code Size Estimations using Function Points	16
5.1.1 Unadjusted Function Points	16
5.1.2 Adjusted Function Points	27
5.1.3 Lines of Code	28
5.2 Efforts, Duration and Team Size Estimation.....	29
5.2.1 Distribution of Effort	29
5.3 Cost Estimates	30
6 Production Checklist	31
7 Best Practice Checklist	32
8 Risk Management	33

9	Quality Assurance	35
10	Monitoring & Control.....	36

1. Introduction

1.1 Project Overview

Wanderers is a web application designed to streamline group travel planning and expense management. Serving as a collaborative platform, Wanderers enables users to organize trips, coordinate itineraries, track shared expenses, and manage travel logistics seamlessly. The platform allows users to create and join trips, invite fellow travelers, and contribute to a shared itinerary in real-time. Additionally, Wanderers features an intuitive expense-splitting system to fairly distribute costs among group members, reducing the hassle of manual calculations. By integrating real-time updates and a user-friendly interface, Wanderers ensures a seamless and efficient travel planning experience.

1.2 Project Description and Scope

Wanderers is a collaborative travel platform designed to simplify the process of planning and managing group trips. Our objective is to develop a comprehensive trip coordination system that allows users to seamlessly organize itineraries, split expenses, and coordinate logistics with fellow travelers.

Key Features of Wanderers include:

- **User-Friendly Interface** – A clean and intuitive design that enables users to easily create, manage, and navigate their travel plans.
- **Trip Organization & Itinerary Management** – Users can create trips, invite participants, and collaboratively build an itinerary, including accommodation details, activity schedules, and travel routes.
- **Expense Tracking & Cost Splitting** – A transparent cost-sharing system that allows users to log expenses, split costs fairly among group members, and track payments in real time.
- **Integrated Communication Tools** – Built-in messaging and notifications to ensure smooth coordination between trip members.

2. Project Organization

2.1 Team Structure

The following is the list of executive roles, as required by CMM level 3.

- Senior Management: Kim Seojin
- Software Configuration Manager: Isaac Chun Jun Heng
- Software Engineering Project Group: Yu Zi Hao Albert, J'sen Ong Jia Xuan, Hazim Khoiruddin, Leong Mininn Miko
- Software Quality Assurance Engineer: Raghav Rajendran Nair
- Representative of Customer: Isaac Chun Jun Heng

2.2 Roles and Responsibilities

Project Manager: Kim Seojin

- Oversees project progress and ensures milestones are met.
- Approves and executes the project plan.
- Assigns tasks and communicates project status to team members.
- Manages and motivates the team to maintain productivity and morale.
- Represents the team in external communications and stakeholder meetings.

Front-end Developer: J'sen Ong Jia Xuan, Leong Mininn Miko

- Implements the user interface for the Wanderers platform.
- Integrates front-end components with the back-end system to ensure seamless functionality.
- Utilizes front-end technologies to develop responsive and interactive travel planning features.
- Conducts usability testing to enhance user experience and interface design.

Back-end Developer: Hazim Khoiruddin, Yu Zi Hao Albert

- Develops and maintains the back-end functionality of the Wanderers system.
- Implements server-side logic to support trip coordination, expense tracking, and user authentication.
- Integrates external APIs and databases for real-time travel data and payment processing.
- Works closely with front-end developers to ensure smooth system communication and data flow.

Release Manager & Quality Assurance Manager: Isaac Chun Jun Heng

- Oversees the release process, ensuring smooth and efficient deployment of updates and new features.
- Manages release schedules and collaborates with the development team to prioritize and coordinate releases.
- Conducts risk assessments and ensures compliance with quality assurance standards.
- Oversees testing strategies to verify software reliability and performance before release.

Release Engineer & Quality Assurance Engineer: Raghav Rajendran Nair

- Implements and maintains the release pipeline, ensuring seamless integration and deployment of system updates.
- Develops automation tools to streamline testing and deployment processes.
- Executes test plans, identifies bugs, and ensures software meets the required quality standards.
- Works closely with developers to troubleshoot issues and optimize system performance.

2.3 Team Communication

Our communication channels include the following:

- Fortnightly meetings are held for the whole team on Tuesdays, 8pm.
- Weekly meetings are held for the developers on Sundays, 10pm.
- Group announcements and updates are sent through our Telegram group.
- Ad-hoc zoom discussions are held as necessary.
- Split up into subgroups as necessary, to work more co-operatively on specific problems
 - Each person owns a particular capability, such that they are the domain expert of that feature and no overlap in responsibilities allows easy finding of individual to consult + more efficient dev process.

3. Process Definition

3.1 Life Cycle Model

The Wanderers project intends to follow the Incremental Development Model, which allows for greater flexibility compared to the traditional Waterfall SDLC. This approach ensures that development progresses through repeated iterations, involving design, coding, unit testing, integration, and quality assurance. Given the time constraints for delivering Wanderers, a Waterfall approach would not be suitable due to its rigid sequential structure.

Methodologies such as Spiral have been avoided due to concerns over the short timeline. Revisiting design procedures extensively within the initial release phase could risk exceeding the project's critical deadlines.

The first iteration of Wanderers will be delivered according to the System Delivery date specified in the Estimation section of this document. Following further client feedback, additional iterations will be planned as needed to refine functionality.

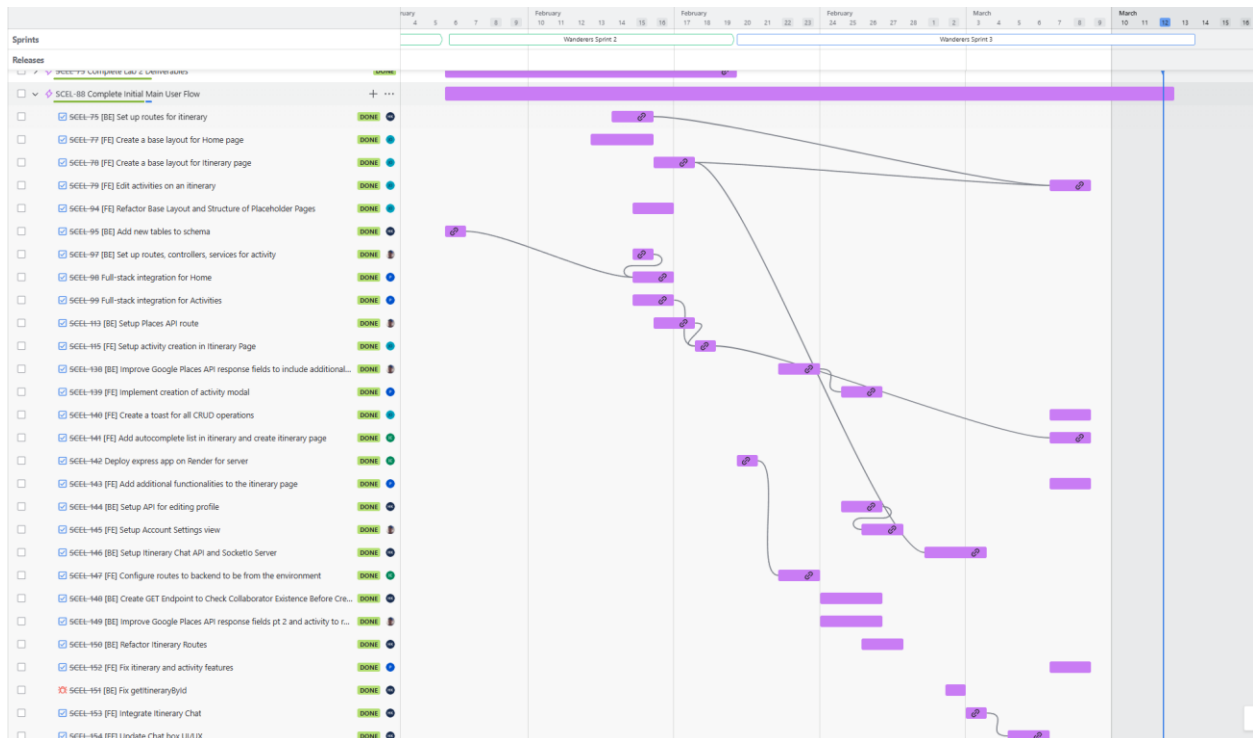
Since Wanderers focuses on collaborative travel planning and expense management, iterative development cycles incorporating user feedback and live testing will help optimize the system, ensuring an intuitive and efficient experience for users.

4. Schedule

4.1 Activity Dependencies and Schedule

Wanderers follows an epic and sprint system, which means that our tasks are planned per sprint and all tickets are in the backlog for grooming/planning. We do both planning in the sprint and planning for items that are in the backlog.

The following image is an example of the timeline planned for our recent sprint. The grey line indicates the dependencies between tasks, which are clearly defined when planning the tasks for each sprint. For ad hoc tasks, if there are dependencies, they are mentioned to the team for re-planning.



In addition, items are tracked in the sprint using the board view, where the details of each task is displayed with the following details indicated by the small icons

1. Deployment status of the branch (every branch is deployed for preview of the application specific to that branch changes)
2. Story point of the task
3. Priority of the task
4. Whether a branch was created + is PR created for a ticket.

Projects

Wanderers ...

Summary Timeline Backlog **Board** Reports List Forms Goals Issues Development Code Security Releases Deployments Archived issues

Q Search board

IC RN MK P +2 Epic Type

TO DO 5

Review 1st Draft of Risk Management Plan Document

COMPLETE LAB 3 DELIVERABLES

04 Mar 2025

✓ SCEL-134 = KJ

Review 1st Draft of Project Plan Document

COMPLETE LAB 3 DELIVERABLES

04 Mar 2025

✓ SCEL-133 = RN

[FE] Implement Expense Calculation and Splitting

✓ SCEL-158 = P

[FE] Itinerary add new collaborator

✓ SCEL-160 = JG

[FE] Update Activity if user is owner or collaborator

✓ SCEL-161 = JG

+ Create issue

IN PROGRESS 1

[Lab 3] Project Estimates --> Project Plan Document

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-127 4 = RN

IN REVIEW 15

[Lab 3] Product Checklist --> Project Plan Document

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-128 1 = RN

[Lab 3] Schedule --> Project Plan Document

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-126 3 = RN

[Lab 3] Best Practice Checklist --> Project Plan Document

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-129 1 = RN

[Lab 3] Complete In-Lab Meeting Minutes

COMPLETE LAB 3 DELIVERABLES

20 Feb 2025

✓ SCEL-120 1 = RN

[FE] Add a time feature for Activity

COMPLETE INITIAL MAIN USER FLOW

12 Mar 2025

✓ SCEL-157 1 11 ^ P

[FE] Implement Activity Sequence Feature

COMPLETE INITIAL MAIN USER FLOW

12 Mar 2025

✓ SCEL-156 0.5 ^ P

[Lab 3] Risk Assessment Document --> 1st Half

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-121 3 = KJ

[Lab 3] Risk Management --> Project Plan Document

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-130 1 = RN

DONE 26 ✓

[FE] Edit activities on an itinerary

COMPLETE INITIAL MAIN USER FLOW

08 Mar 2025

✓ SCEL-179 ✓ = JG

[FE] Implement API Calling for Itinerary data

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-116 ✓ 5 = P

[FE] Setup confirm account, forget password & reset password views

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-136 ✓ 0.5 = JG

[BE] Improve Google Places API response fields to include additional data

COMPLETE INITIAL MAIN USER FLOW

23 Feb 2025

✓ SCEL-138 ✓ 0.5 = JG

[BE] Setup confirm account, forget password & reset password routes, controller

COMPLETE LAB 3 DELIVERABLES

✓ SCEL-137 ✓ 0.5 = JG

[FE] Implement creation of activity modal

COMPLETE INITIAL MAIN USER FLOW

26 Feb 2025

✓ SCEL-139 ✓ 2 = P

[FE] Create a toast for all CRUD operations

COMPLETE INITIAL MAIN USER FLOW

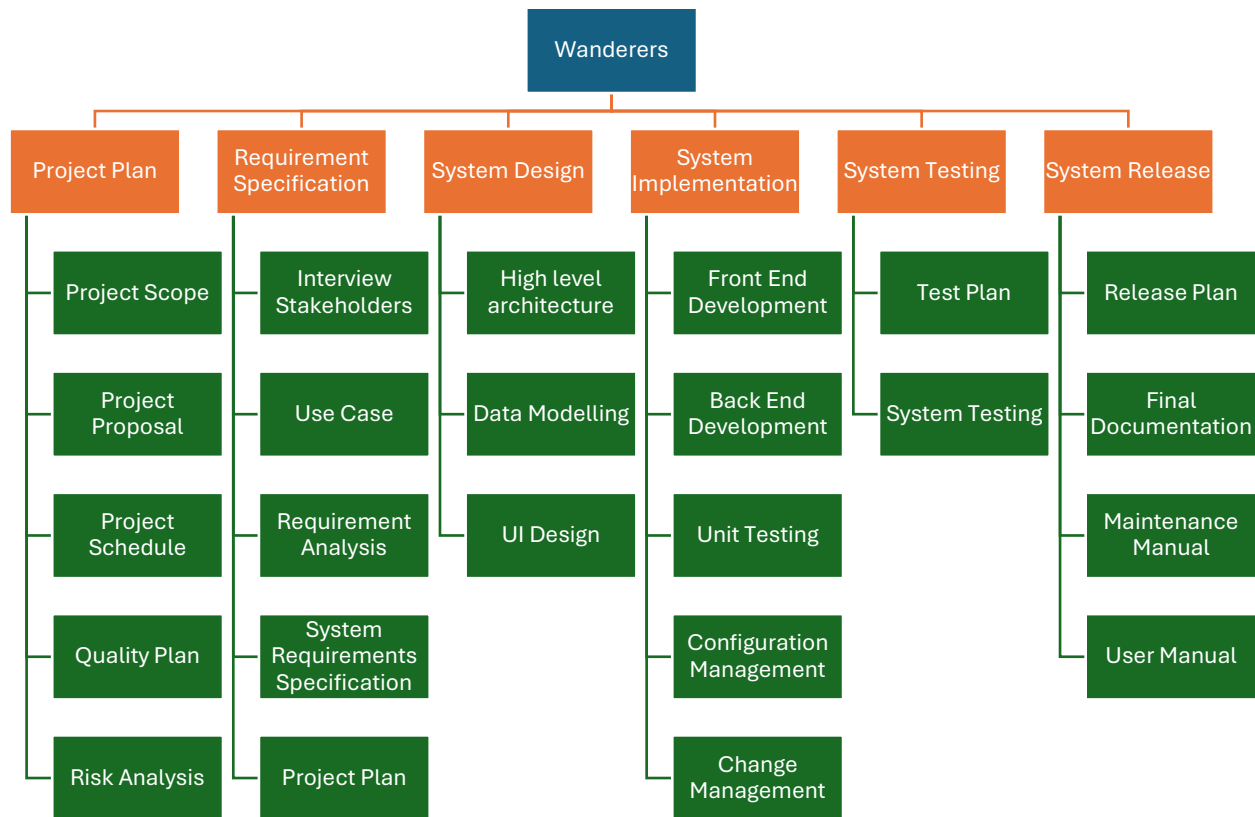
08 Mar 2025

✓ SCEL-140 ✓ 2 = JG

[FE] Add autocomplete list in itinerary and create itinerary

WONT DO ✓

4.2 Work Breakdown Structure



4.3 Work Packages

The entire project work is broken down by the important phases of the software development life cycle. They include the following:

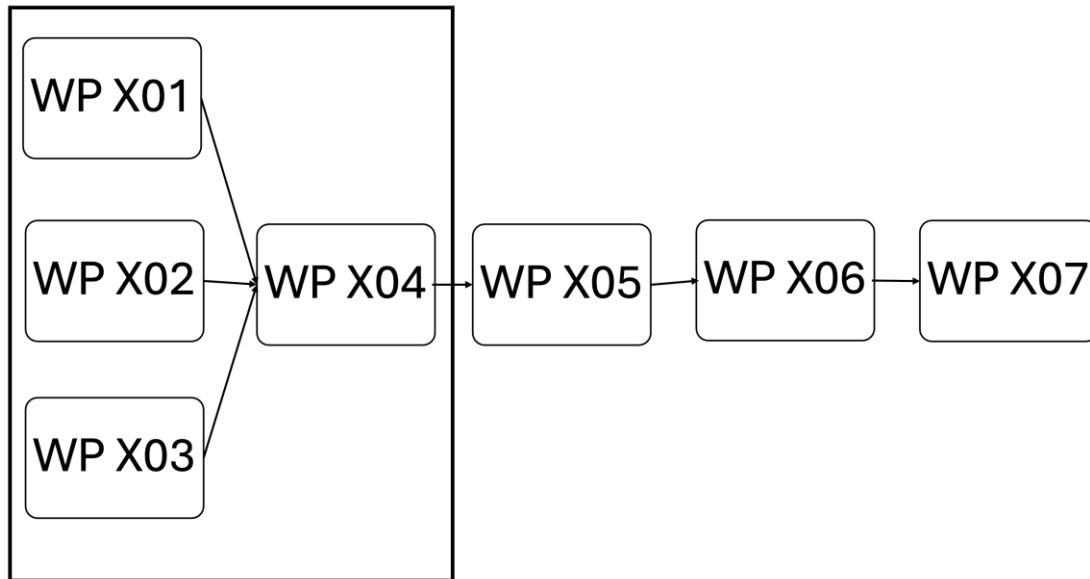
1. Project Plan
2. Requirement Specification
3. User Interface
4. Technical Architecture
5. Data Modeling
6. Coding & Unit Testing
7. Integration & Quality Assurance

4.4 Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

Work Package #	Work Package Description	Duration	Dependencies
X01	Project Plan	7 days	--
X02	Requirement Specification	7 days	--
X03	User Interface	7 days	--
X04	Technical Architecture	12.1 days	X01, X02, X03
X05	Data Modeling	7 days	X04
X06	Coding & Unit Testing	16.2 days	X05
X07	Integration & System Testing	16.2 days	X06

The following Activity Network Diagram describes the above in more graphical detail:



Note that work package X05 is dependent on all work packages encapsulated by the larger boxes linked to its left. For instance, WP X05 may not start until WP X01- X04 has been finished.

4.5 Work Package Details

Work packages are listed below. A team member, indicated in bold, has been assigned as primarily responsible for each work package and will coordinate that package.

Project	Wanderers
Work Package	X01— Project Plan (1 of 7)
Assigned To	Raghav , Isaac, Seo Jin, J’sen, Albert, Hazim, Miko
Effort	7PD
Start Date	Thursday, 23/01/2025
Purpose	To determine an introductory overview of the project, to be refined in later work packages.
Inputs	None
Activities	This work package includes providing a brief overview of the project, its objectives, and a set of proposed project deliverables throughout the development of the software cycle. The people responsible for this work package will also be transcribing ideas brought up in the group meeting discussion into a formal report
Outputs	A written document of the Project Plan Introduction.

Project	Wanderers
Work Package	X02— Requirement Specification (2 of 7)
Assigned To	Seojin , Isaac, Raghav, J’sen, Albert, Hazim, Miko
Effort	7PD
Start Date	Thursday 06/02/25
Purpose	To establish a common understanding between the customer and the software project team of the customers’ requirements to be addressed by the project
Inputs	Customer’s requirements
Activities	Identify “the customer”, interview customer, write and inspect customer requirement and build requirements.
Outputs	A written document of the requirement specification

Project	Wanderers
Work Package	X03— User Interface (3 of 7)
Assigned To	J’sen, Raghav, Isaac, Seo Jin, Albert, Hazim, Miko
Effort	7PD
Start Date	Thursday 06/02/25
Purpose	To build the user interface between the system and the customer, to make it easy use, and friendly to the customer
Inputs	User Information
Activities	To get the user information, user request, display the dialog between system and user, display the result of request
Outputs	User Interface

Project	Wanderers
Work Package	X04— Technical Architecture (4 of 7)
Assigned To	Albert , J’sen, Raghav, Isaac, Seo Jin, Hazim, Miko
Effort	12.1PD
Start Date	Thursday 06/02/25
Purpose	To do the high level architecture design
Inputs	Project Plan Work Packages (X01 to X03 inclusive).
Activities	
Outputs	High Level Design and Architectural Specification.

Project	Wanderers
Work Package	X05— Data Modeling (5 of 7)
Assigned To	Hazim , J’sen, Raghav, Isaac, Seo Jin, Albert, Miko
Effort	7PD
Start Date	Thursday 20/02/25

Purpose	To build the project's database
Inputs	Project Plan Work Packages (X01 to X05 inclusive)
Activities	Analyze the data flow relationships, entity relationships
Outputs	A written document of the data modeling

Project	Wanderers
Work Package	X06— Coding & Unit testing (6 of 7)
Assigned To	Isaac, J'sen, Raghav, Hazim, Seo Jin, Albert, Miko
Effort	16.2PD
Start Date	Thursday 20/03/25
Purpose	To implement the system as per the requirements specification and other associated documents. This work package includes such additional activities as preliminary unit testing.
Inputs	Project Plan Work Package X06.
Activities	Programmers will implement the modules according to the design specifications noted in the Specification document.
Outputs	Source code and header files

Project	Wanderers
Work Package	X07— Integration & System Testing (7 of 7)
Assigned To	Isaac, J'sen, Raghav, Hazim, Seo Jin, Albert, Miko
Effort	16.2PD
Start Date	Thursday 13/03/25
Purpose	To identify and fix logical and syntactical errors produced during the implementation of the System, and setting up drivers and stubs to see how the module responds to various inputs. Black box testing as well as white box testing might be conducted to check for logical errors. All the testing procedures will be documented in the Test Plan report. If problems are found, they will be noted and fixed at the earliest possible time.
Inputs	Project Plan Work Package X07
Activities	The Integration testing team may try to simulate how a user might interact with the system. Similar to Unit Testing, Integration Testing may require the development of stubs and drivers as well, but here this is more geared towards the higher (overall system) level. Testers may also examine issues such as system performance and integrity. Heuristics assessment plays an important role in this work package, as intelligence components will define eventual system success.
Outputs	Test Report

5 Project Estimates

5.1 Code Size Estimations using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

5.1.1 Unadjusted Function Points

Wanderers supports the following proposed functions, as detailed in the Functionalities section of the System Requirements Specification (SRS) Document.

Wanderers' Functionalities for users:

1. Register Account
2. Log in
3. Log out
4. Reset or Change Password
5. Profile Management
6. Trip Planning
7. Cost Splitting
8. Discussion & Communication

The assessment of unadjusted function points is determined through five core components of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. The complexity and measure of each element ranges from Low Complexity, Medium Complexity and High Complexity. The complexity estimation uses a matrix based on the number of File Type Referenced (FTRs)/Record Element Types (RETs) against the number of Data Elements (DEs).

Legend:

1. File Types Referenced – number of logical files referenced by the functionality
2. Data Elements – number of data elements (fields or attributes involved in the functionality)
 - a. e.g Register Account involves fields like **email, password**

Rating Inputs:

- Application oriented data (such as file names and menu selections, code written to handle the data)

Matrix:

Files Type Referenced (FTR)	Data Elements		
	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

Complexity Estimation:

Functionality	FTR/DE Count	Description	Complexity		
			L	A	H
Register Account	FTR: 3	The system interacts with the backend for sending a request to the Supabase database to store the user details (username, email address, password) and sends an email via the SMTP service for confirmation			✓
	DE: 3	The user inputs username, email address and a password . The password has multiple criteria for fulfilment, but counts as one field.			
Login	FTR: 2	The system interacts with the backend hosted in Express and crosschecks against the Supabase database for verifying of user credentials.		✓	
	DE: 2	The user inputs email address and password to authenticate and gain access to the system			
Logout	FTR :2	The system interacts with the backend hosted in Express and terminates the session token in the Supabase database.	✓		

	DE: 1	The user confirmation (clicking of log out button counts) as the data element.			
Reset or Change Password	FTR: 3	The system interacts with the backend hosted in Express for reset of the password, which then interacts with the SMTP service to send a password reset email with token.		✓	
	DE: 3	User provides their email address , and after clicking the email, they are prompted to key in their new password and confirm password.			
Profile Management	FTR: 3	The system interacts with the backend hosted in Express in profile management, which can either interact with the S3 Object Storage for acquiring of image data, or the normal Supabase database for changing of personal details.		✓	
	DE: 3	The user modifies personal details like their profile photo, current password, profile description			
Trip Planning	FTR: 3	The system interacts with backend hosted in Express and Supabase storage for storing of information. Also, SMTP service is used when inviting other users to collaborate on the itinerary through sending them a link to their email.			✓
	DE: 10+	User inputs details such as destination, dates, collaborators, name, and other trip-related information			
Cost Splitting	FTR: 3	System interacts with backend hosted in Express to add a cost splitting log in the Supabase database and then display it on the user interface.			✓

	DE: 4	User provides expense details, participant names, individual cost contributions and total cost splitting based on the input.			
Discussing and Communication	FTR: 2	System interacts with backend socket.io for sending of images to one another , and then display it on the user interface		✓	
	DE: 1	The user enters the message content into the chatbox, which sends a message to all the users in the itinerary			
Count			1	5	2

Rating Outputs:

- Transaction data output to the user or underlying process such as tokens.

Matrix:

Files Type Referenced (FTR)	Data Elements		
	1-4	6-19	Greater than 15
Less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 2	Average (5)	High (7)	High (7)

Complexity Estimation:

Functionality	FTR/DE Count	Description	Complexity		
			L	A	H
Register Account	FTR: 3	The system interacts with the backend hosted in Express for querying the Supabase database and the SMTP Service for sending of confirmation email.		✓	
	DE: 3	System might output the success or error message regarding account creation, with the verification instructions .			
Login	FTR: 2	After login, system outputs success message and authentication token		✓	
	DE: 2	The output includes the authentication token or error message depending on the success or failure of the login process.			
Logout	FTR: 2	System interacts with backend hosted in Express to terminate the session data associated with the user in the Supabase database	✓		
	DE: 1	Output consists of a simple confirmation message			

		indicating that the user has been logged out successfully			
Reset or Change Password	FTR: 3	System interacts with backend hosted in Express to terminate the session data associated with the user in the Supabase database, alongside the SMTP service to send a reset password link.		✓	
	DE: 2	The output consists of a simple confirmation message indicating that the password has been updated or an error message if the reset or change process failed.			
Profile Management	FTR: 2	System interacts with backend hosted in Express to update the profile details in the Supabase database.		✓	
	DE: 2	The output includes a success message when profile is updated successfully or an error message if the operation fails.			
Trip Planning	FTR: 4	System interacts with backend hosted in Express, Supabase database storage and the S3 Object Storage to acquire details about the trip. The system also interacts with the Google Places API for querying of place details.			✓
	DE: 10+	The output includes trip details such as trip name, trip timing, and other trip related information.			
Cost Splitting	FTR: 2	System interacts with backend hosted in Express, Supabase database storage for acquiring the cost splitting details			✓
	DE: 4	The output includes a detailed breakdown of the cost for each participant and the total			

		amount to be paid by each person			
Discussing and Communication	FTR: 2	System interacts with backend socket.io for sending of images to one another , and then display it on the user interface		✓	
	DE: 1	The output consists of messages in the chatbox from other users			
Count			1	5	2

Rating Inquires:

- Interactive inputs requiring a response (not update file)
- Database authentication, validation are part of this section.

Matrix:

Files Type Referenced (FTR)	Data Elements		
	1-4	6-19	Greater than 15
Less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 2	Average (5)	High (7)	High (7)

Complexity Estimation:

Functionality	FTR/DE Count	Description	Complexity		
			L	A	H
Register Account	FTR: 1	The system queries the Supabase User Database to check if the email or username already exists.		✓	
	DE: 2	The user inputs email and username to check if they are available.			
Login	FTR: 1	The system queries the Supabase User Database to validate the username/email and password, ensuring that the credentials are correct.		✓	
	DE: 2	The user inputs username and password to authenticate and log in.			
Logout	FTR: 1	System interacts with backend hosted in Express to terminate the session data associated with the user in the Supabase database	✓		
	DE: 1	The user clicks on the log out button, which triggers a session termination.			

Reset or Change Password	FTR: 1	The system queries the Supabase User Database to check if the email is valid before initiating the reset process.		✓	
	DE: 2	The user enters email and new password to initiate the reset process.			
Profile Management	FTR: 1	The system queries the Supabase User Database to retrieve and display the user's current profile information (name, email, preferences, etc.).		✓	
	DE: 2	The user goes to their own profile and edit personal details such as profile photo and description.			
Trip Planning	FTR: 1	The system queries the Supabase Trip Planning Database to retrieve trip data for that itinerary with specific id			✓
	DE: 10+	The user enters trip destination, dates, budget, and participants to create or modify a trip (trip-related operations)			
Cost Splitting	FTR: 1	The system queries the Supabase Cost Splitting Database to retrieve past cost data or calculate the share of costs among participants.			✓
	DE: 4	The user inputs data related to expenses, participants, and any additional parameters to divide the total cost.			
Discussing and Communication	FTR: 1	The system queries the Supabase Chat Database to retrieve previous messages, discussions.		✓	
	DE: 1	The user inputs message content and sends or receives			

		messages in a chat or discussion.			
Count			1	5	2

Rating Logical Files:

- Logical master files in the system

Matrix:

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low (7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Complexity Estimation:

Functionality	Complexity		
	L	A	H
User Account Management (includes Login, Logout, Register, Reset Password)		✓	
Trip Planning (Create, read, edit update itineraries, create read edit update activities)			✓
Discussion and Communication (real time chat between users)	✓		
Cost Splitting		✓	
Count	1	2	1

Rating Interfaces:

- Machine readable interfaces to other systems
- This includes APIs used etc.

Matrix:

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low (7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Complexity Estimation:

Functionality	Complexity		
	L	A	H
Google Places API	✓		
SMTP Service	✓		
Count	2	0	0

Calculation of Unadjusted Function Points:

Characteristic	Low		Medium		High	
Inputs	1	× 3	5	× 4	2	× 6
Outputs	1	× 4	5	× 5	2	× 7
Inquiries	1	× 3	5	× 4	2	× 6
Logical Files	1	× 7	2	× 10	1	× 15
Interfaces	2	× 5	0	× 7	0	× 10
Unadjusted FP	27		85		53	
Total=L+M+H	165					

5.1.2 Adjusted Function Points

Scoring (0 – 5)
0 = No influence
1 = Insignificant influence
2 = Moderate influence
3 = Average influence
4 = Significant influence
5 = Strong influence

Influence Factors	Score	Detail
Data Communications	3	The application is not only a front-end and involves a backend, plus interactions with external APIs for authentication and data retrieval, which requires a moderate level of data communication and dependencies.
Distributed Functions	2	Some functionalities require interaction with distributed services like APIs, and main part of our processing is on the backend
Performance	3	Response time or throughput is critical during all business hours. No special design for CPU utilization was required. Processing deadline requirements with interfacing systems are constraining.
Heavily used	2	Some security or timing considerations are included.
Transaction rate	2	Regular transactions such as planning and editing of itinerary/activities should occur. Daily peak transaction period is anticipated at any hour of the day, specifically at night as majority of people are free at night (prediction)
On-line data entry	3	Multiple of our functionalities involve online data entry, such as registering, resetting passwords, login, editing and creating itineraries and activities.
End-user efficiency	3	The user interface and experience is heavily focused on as it is part of our project goals to make the UI responsive and easy to use.
On-line data update	2	Regular updates are likely and frequent due to online data entry updates such as operations on the itineraries and activities.
Complex processing	2	Interactions with external APIs such as Google Places API and real-time data handling suggest a moderate level of complexity.

Reusability	1	No special considerations were stated by the user.
Installation Ease	0	The application is web based, which means that there is little to no need for installation, and just using a web browser will do.
Operational Ease	3	Operational ease is to be a priority for this project as the focus is on user efficiency and quick operations and efficient handling.
Multiple sites	1	Since our application is hosted on the web, it should be naturally accessible from various locations
Facilitate change	3	The system is very likely to be facilitated to change, considering the modern web-based environment (Next.js) and interactions with our backend and databases that suggest the system would be updated frequently, which implies our system must be adaptable.
Total score	30	
Influence Multiplier = Total score \times 0.01 + 0.65 = 30 \times 0.01 + 0.65 = 0.95		
Adjusted FP = Unadjusted FP \times Influence Multiplier = 165 \times 0.95= 155.1		

5.1.3 Lines of Code

According to QSM Function Points Languages Table, each Function Point requires 47 lines of code if the application is implemented using Javascript/Typescript.

Reference: <https://www.qsm.com/resources/function-point-languages-table>

Hence, **Lines of Code** = 155.1 FP \times 47 LOC/FP = **~7290 LOC**

5.2 Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use function points as the basis to calculate Effort, Duration, Team size and finally the schedule. The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations.

- From industry benchmarks. Production rate is 62 LOC / PD
- Working days include 5 days in a week.
- Effort = Size / Production Rate = (7290 LOC) / (62 LOC/PD)₁ = **117.5 PD**
- Duration = $3 \times (\text{Effort})^{1/3} = 3 \times (117.5)^{1/3} = \mathbf{14.7 \text{ Days}}$
- Initial schedule = 14.7 Days / 5 days a week = **2.94 Weeks**
- Team size = 117.5 PD / 14.7D = 7.99P = **8 Persons**
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 117.5 PD × 8 hours = **940 PH**

From the calculations, since the team only has a team size of 7, we must adjust the duration of the work as that takes into accounts of 8 persons. After adjustment, the duration would be **3.36 weeks**. The calculation is as follows:

- Duration = Effort/Team Size = 117.5 PD / 7 = 16.79 days (naively)
- Duration in Weeks = 16.79 / 5 days a week = **3.36 weeks**.

5.2.1 Distribution of Effort

1990's Industry Data	Work Package	Distribution	Estimate s
Preliminary Design (18 %)	Project Plan	9%	84.6
	Requirement Specification	9%	84.6
Detailed Design (25 %)	User Interface	7%	65.8
	Technical Architecture	11%	103.4
	Data Modeling	7%	65.8
Code & Unit Testing (26 %)	Code & Unit testing	21%	197.4
	Online Documentation	5%	47
Integration & Test (31 %)	Integration & Quality Assurance	31%	291.4
	Extrapolated total effort		940
	2% for project management		18.8

	3% for contingency		28.2
	Total effort		987

These duration estimates assume that each team member works an equal amount on any given work package.

¹ Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 31 LOC/PD for United States; 62 LOC/PD for Canada

5.3 Cost Estimates

Hardware:

Developer Workstations:

7 x Personal laptops	Total: \$0.00
----------------------	---------------

Software:

Free license-based Software:

GitHub	\$0.00
Google Places API	\$0.00
SMTP Service	\$0.00

Software License Provided by Third Party:

Microsoft 365 Apps	\$0.00
Canva (for presentations)	\$0.00
Visual Paradigm	\$0.00

Other Resources:

Staff:

7 employees with 987 working hours at \$18.00/hour	\$17,766
----------------------------------------------------	----------

Wanderers' hardware and software requirements and responsibilities extend only to the development requirements of the team to complete the assigned project, which is

described in the Introduction section of this document. Wanderers team will also demonstrate the completed product.

6 Production Checklist

The plan is that the items listed below will be delivered on the stated deadlines

Deliverable	Estimated Completion Date	Final Deadline
In-lab meeting minutes	Every lab session	
Post-lab meeting minutes	Every meeting	
Backlog	Before every lab session	
Project Proposal	04/02/2025	06/02/2025
Use Case Model	04/02/2025	06/02/2025
Software Requirement Specification	18/02/2025	20/02/2025
Quality Plan	18/02/2025	20/02/2025
Project Plan	12/03/2025	13/03/2025
Risk Management	12/03/2025	13/03/2025
Prototype Code	12/03/2025	13/03/2025
Prototype Documents	12/03/2025	13/03/2025
Design Report on Software Maintainability	25/03/2025	27/03/2025
Configuration Management Plan	25/03/2025	27/03/2025
Change Management Plan	25/03/2025	27/03/2025
Release Plan	25/03/2025	27/03/2025
Presentation Slides	25/03/2025	27/03/2025
Test Plan	25/03/2025	27/03/2025
Test Cases & Requirements Report	25/03/2025	27/03/2025
Final Documentation	25/03/2025	27/03/2025

7 Best Practice Checklist

Practice

Document what we do; all documentation must be in a standardized format.

Pay attention to requirements, check for ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification.

Keep it simple. Complexity management is one of the major challenges. Strive to:

- Minimize interfaces between modules, procedures and data.
- Minimize interfaces between people, otherwise exponential communication cost
- Avoid fancy product functions, design as long as the functionality meets the customer requirements

Require Visibility. We must see what we build otherwise we can measure the progress and take management action. This includes: the manager must have good communication with his or her employees; require developers to make code available for review; review design for appropriateness.

Plan for continuous change. We must:

- All manuals design, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes
- New revisions should be approved before being made and checked for quality and compliance after being made
- Use a configuration management system and make processes
- Required maintenance

Don't underestimate. We must be careful to obtain accurate estimates for: time, effort, overhead, meeting time, and especially effort on integration, testing, documentation and maintenance.

Code reviews are a much more efficient method to find software defects. Plan and manage code reviews between team members

Software testing will use both black box and white box testing. It will involve unit, functional, integrating and acceptance testing.

8 Risk Management

Besides general risk management practices, the following risks have been identified for the **Wanderers** project:

More changes to requirements than anticipated

Impact Severity: High

Probability: 25%

Impacts: Depending on the stage at which changes occur, this could range from needing to update the requirements documentation to requiring a complete redesign of core functionalities.

Risk Reduction: Conduct thorough requirement gathering sessions early on. Maintain clear communication with stakeholders and set expectations regarding the impact of requirement changes.

Delays in system integration

Impact Severity: High

Probability: 20%

Impacts: Delays in integrating front-end and back-end components could push the entire project timeline back, affecting deployment.

Risk Reduction: Establish clear integration milestones and conduct regular testing to ensure compatibility between system components.

Underestimation of system complexity

Impact Severity: Moderate

Probability: 30%

Impacts: If the system complexity is greater than initially estimated, additional time and resources may be needed, potentially affecting the project schedule.

Risk Reduction: Continuously update system estimates as development progresses. Conduct early-stage prototyping and feasibility checks to identify potential complexities.

Team members leaving before project completion

Impact Severity: Extreme

Probability: 1%

Impacts: Increased workload on remaining team members, potential loss of specialized knowledge, and slower project progress.

Risk Reduction: Encourage documentation of all processes and decisions. Foster a collaborative team environment and consider knowledge-sharing sessions to mitigate the risk of losing critical expertise.

Coordination challenges within the team

Impact Severity: Moderate

Probability: 40%

Impacts: Miscommunication or lack of coordination could lead to misunderstandings, duplication of effort, and missed deadlines.

Risk Reduction: Follow a structured communication plan with regular check-ins. Use project management tools like Jira and Slack to ensure transparency in task assignments and updates.

Customer disengagement or project cancellation

Impact Severity: Extreme

Probability: 1%

Impacts: If the customer loses interest or cancels the project, all work done up to that point may be wasted.

Risk Reduction: Maintain close communication with stakeholders, provide regular progress updates, and ensure that the project aligns with their expectations and needs.

By proactively identifying and mitigating these risks, the **Wanderers** team aims to ensure the successful completion and deployment of the project.

9 Quality Assurance

The Wanderers project will achieve quality assurance by adhering to the standards set by the team. Specific procedures and details will be outlined in the Quality Plan.

To ensure a high level of software quality, the project will also employ the following testing methodologies:

- **Unit Testing** – Individual system components will be tested in isolation to verify their correctness.
- **Integration Testing** – Testing the interaction between front-end and back-end components to ensure seamless communication.
- **System Testing** – The entire system will be tested as a unit to ensure that all components function correctly together.
- **Load Testing** – The system will be tested under high user traffic conditions to evaluate performance and scalability.

These methodologies will be used to test key aspects of Wanderers:

- **System Functionality** – Ensuring that all features work as intended and that software flaws are eliminated.
- **Data Accuracy & Consistency** – Validating that expense tracking, itinerary planning, and group management features store and process data correctly.
- **User Experience & Responsiveness** – Conducting usability testing to ensure the platform is intuitive, user-friendly, and responsive across different devices.

The Wanderers team will utilize realistic test cases based on actual travel planning scenarios to validate system behaviour. Additionally, extreme cases (such as large group travel plans with complex cost-sharing) will be tested to ensure the platform can handle diverse user needs effectively.

10 Monitoring & Control

Effective monitoring and control procedures are essential for ensuring the successful progress of the Wanderers project. The key strategies include:

Quantitative Measurement of Resource Allocation – Tracking resource utilization, including human effort and development time, to ensure efficient progress toward project milestones. Task effort estimates outlined in this document allow for accurate progress tracking and resource reallocation when necessary.

Identification of Major Project Risks – Early identification of potential risks enables the implementation of mitigation strategies before they impact project delivery. Key risks and their preventive measures are outlined in the **Risk Management** section.

Regular Progress Reviews – The team will hold weekly meetings to assess the status of all project tasks, including development, testing, and deployment. These reviews will ensure that deliverables remain on schedule and address any blockers promptly.

Timeline Planning and Task Decomposition – The Wanderers project timeline is structured into well-defined milestones and subcomponents. This hierarchical task breakdown facilitates accurate scheduling, workload distribution, and continuous progress assessment. Project subcomponents and timeline estimates are detailed in the **Work Breakdown Structure** section of this document.

These monitoring and control strategies will help maintain alignment with project goals, ensuring timely delivery and a high-quality product.