HelpMe

Project Plan
Version 1.0

By: Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	Oct 4th, 2001	Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee	First version

Table of Contents

1 Introduction

1.1 Project Overview

The HelpMe project is an android base application that acts as a platform that connects users who are looking for specific services on short notice and users who are able to provide these services on short notice. Given a service's category or region, HelpMe will be able to sort and display services offered by others that are of the customer's requirement.

1.2 Project Description and Scope

HelpMe serves as a marketplace for users to post requests and offer services to any other user. As such, HelpMe must be able to fulfil the following:

- Users must be able to create accounts that retain account information such as name and contact information.
- Users can create service requests or offers where other users can respond.
- Service posts should include title, and description in addition to contact information.
- Users must be able to sort and respond to each request.

These items are an example of features that should be available on the application, future improvements or additional features will be discussed in a later document.

HelpMe will only enable users to find and post services. Communication between parties and any monetary transactions are out of its scope and should be done outside of HelpMe application.

The Service posted should include contact details as well as location of service if applicable. This enables efficient sorting of services posted. As such, each service should exist in a specific category.

Think2 has been contracted to fulfil the software. Think2 has been given permission to perform data design and requirements research independently.

The basis of the HelpMe application is an intelligent search facility that takes a number of possible constraints into account. These can be such limits as:

- The user intends to hire a contractor of a certain expertise;
- The user may put up a request for a job;
- The user must be able to choose jobs from contractors of a particular region.

These items are given as an example. Final possible constraints will be discussed in a later document.

Given these preferences, HelpMe retrieves a list of possible services, bearing in mind that the nature of the job and region are extremely important to the user.

As this project involves only HelpMe, such aspects as communication between users and financial transactions are beyond its scope. These are the responsibility of the users themselves. Think2 shall provide the deliverables outlined in this document, along with database schema, a test data set, and all source code.

HelpMe is responsible for the job selection process and putting up of service offers or requests once the users have successfully identified themselves. The agent's responsibilities are

complete once a service has been selected, and both users agree to collaborate on a job.

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

2 Project Organisation

2.1 Roles and Responsibilities

Project Manager: Quek Xuan Hao

- 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

Lead Developer/Back-End Developer: Antoine Tran

- Collects information from client interviews
- Develops concepts of the system for designers
- Creates requirement specification document
- Ensure good communication amongst developers

Release Manager/Engineer: Ong Chun Guan Marcus

- Oversees software delivery life cycle
- Provides feedback and schedule releases
- Assigns tasks and reports status of project to team members
- Manages and motivates team members
- Define strategic release management plans

Back-End Developer: Li Jin Xuan

- Compile and analyse data, processes and codes
- Collaborate with front-end to design cohesive codes
- Ensure smooth transition of data from back to front end
- Implement data storage solutions
- Works closely with and assists lead developer

Front-End Developer: Chen Kian Leong

- Designs User Interface
- Ensures stability and response time of the system meet the requirements
- Creates user manual

Quality Assurance Engineer: Tanya Banrjee

- Ensures acceptable software quality
- Designs testing strategies
- Creates and manages test plan
- Verify software requirements
- Executes test procedures

Quality Assurance Manager: Siti Nur Umm'aira Phang

- Ensures acceptable software quality
- Designs testing strategies
- Creates and manages test plan
- Verify software requirements
- Executes test procedures

2.2 Team Communication

Think2 communication channels include the following:

- Weekly meetings are held on Wednesday online on zoom.
- Group announcements and updates are broadcast in telegram.
- Physical meetings and discussions are held as necessary.
- Split up into subgroups as necessary, in order to work more cooperatively on specific problems.

3 Process Definition

3.1 Lifecycle Model

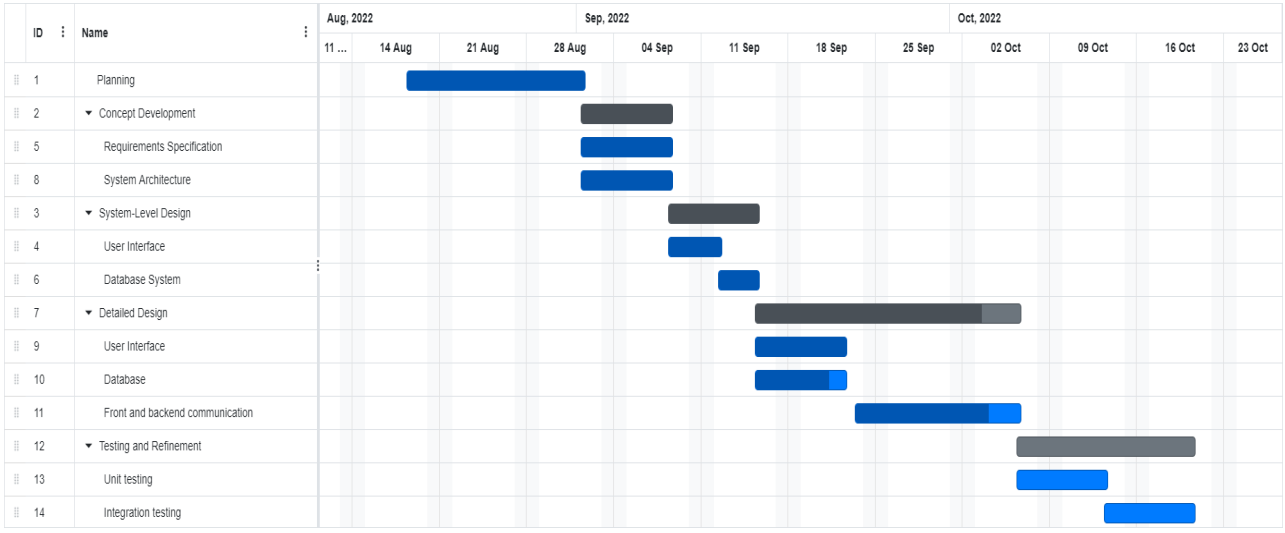
Think2 intends to use the Incremental Development Model throughout the HelpMe project. This methodology is more flexible than the traditional Waterfall SDLC due to repeated iterations involving design, coding, unit testing, integration, and quality assurance. The Waterfall SDLC is not a viable choice due to the short timeline available for the HelpMe project to reach delivery quality.

Think2 has chosen to avoid such methodologies as Spiral because of concerns over the short timeline. Should design procedures, for example, need to be revisited within the first release date, it is likely that the project will overshoot its critical schedule.

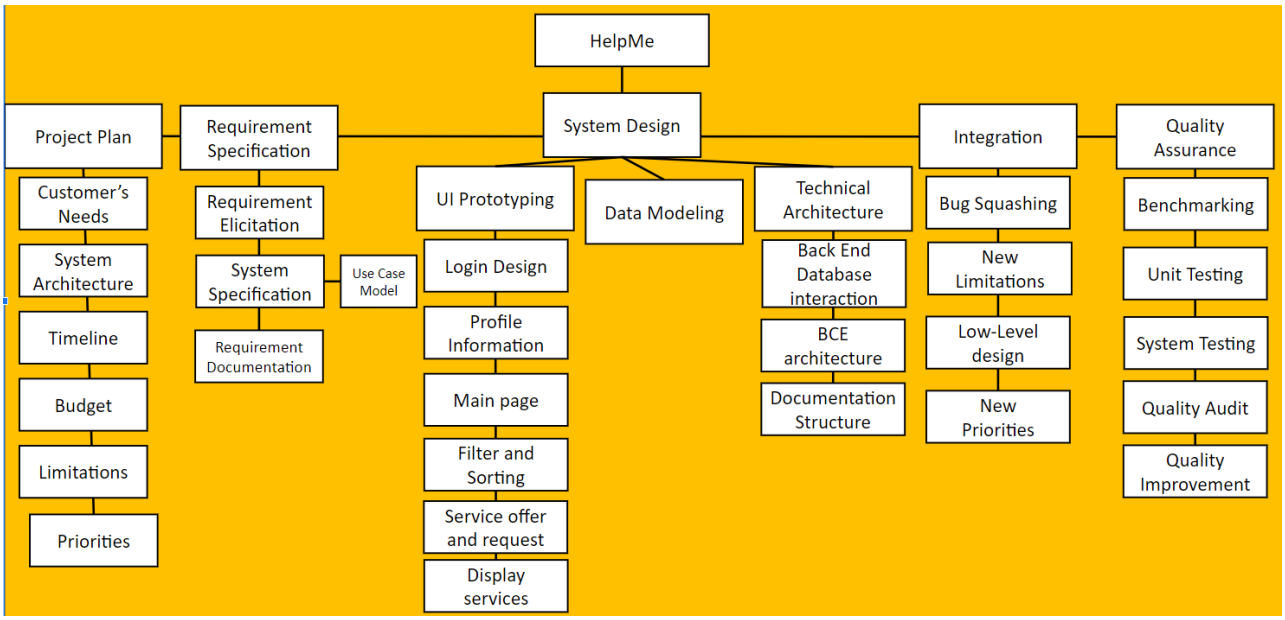
Think2 intends to deliver the first prototype of functionality on the System Delivery date indicated in the Estimations section of this document. After further client interaction, further iterations should occur as necessary. Due to the nature of this application, further iterations may affect users' experience if live testing is done and hence testing shall be done offline such that the application behaves in an accurate and logical manner before update, providing a more effective search experience for users.

4 Schedule

4.1 Activity Dependencies and Schedule



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:

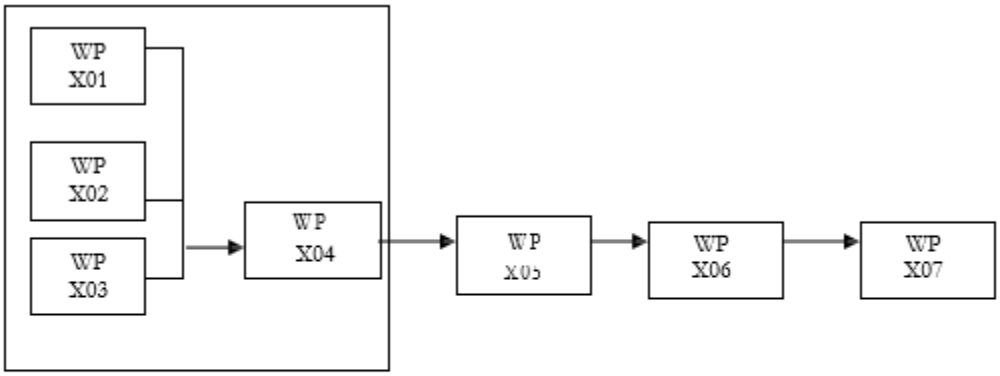
1. Project Plan
2. Requirement Specification
3. User Interface
4. Technical Architecture
5. Data Modelling
6. Coding & Unit Testing
7. Integration & Quality Assurance

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 Modelling	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	Think2 HelpMe Software System
Work Package	X01— Project Plan (1 of 7)
Assigned To	Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee
Effort	7PD
Start Date	Thursday, 15/09/2022
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	Think2 HelpMe Software System
Work Package	X02— System Requirement Specification (2 of 7)
Assigned To	Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee
Effort	7PD
Start Date	Thursday, 08/09/2022
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 requirements and build requirements.
Outputs	A written document of the requirement specification.

Project	Think2 HelpMe Software System
Work Package	X03— User Interface (3 of 7)
Assigned To	Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee
Effort	7PD
Start Date	Thursday, 15/09/2022
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	Think2 HelpMe Software System
Work Package	X04— Technical Architecture (4 of 7)
Assigned To	Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee
Effort	7PD
Start Date	Thursday, 22/09/22
Purpose	To do the high level architecture design
Inputs	Project Plan Work Packages (X01 to X03 inclusive).
Activities	Entails the linkage between software infrastructure such as usage of language, the type of server database. For hardware infrastructure, we will look into the type of system we will be implementing the application on. The general skeleton structure will be the key consideration to prevent major design changes and refactoring in the near future.
Outputs	High Level Design and Architectural Specification

Project	Think2 HelpMe Software System
Work Package	X05— Data Modeling (5 of7)
Assigned To	Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee
Effort	7PD
Start Date	Thursday, 29/09/22
Purpose	To build the project's database
Inputs	Project Plan Work Packages (X01 to X05 inclusive).
Activities	Analyse the data flow relationships, entity relationships Establishing the database while reviewing the class diagram to finalise on the Data Flow Diagram. Data dictionary will be produced which entails the meaning, relationships of one entity to another entity.
Outputs	A written document of the data modelling

Project	Think2 HelpMe Software System
Work Package	X06— Coding & Unit testing (6 of 7)
Assigned To	Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee
Effort	16.2PD
Start Date	Thursday, 29/09/2022
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	Think2 HelpMe Software System
Work Package	X07— Integration & System Testing (7 of 7)
Assigned To	Antoine Tran, Chen Kian Leong, Li Jin Xuan, Ong Chun Guan Marcus, Quek Xuan Hao, Siti Nur Umm'aira Phang, Tanya Banerjee
Effort	16.2PD
Start Date	Thursday, 6/10/2022
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	A test report.

5 Project Estimates

5.1 Code Size Estimation 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

Think2 HelpMe project supports the following proposed functions:

Customer:

- User Account Registration
- User Login
- Mange Profile
- View all services
- View user's own posted services
- Provide filtered results based on user's chosen criteria
- View details about interested services
- Request/Offer services
- Delete service posts

Administrator:

- List customer's service request
- List contractor's service offer

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, 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:

- User Registration (username, contact number, email, password)
- Delete service posts (userID, postID)
- User Login (email, password)
- Reset password (email)
- Modifying user's account information (username, email, contact number, userID)
- Request/Offer Services (title, price, description, offer/request, area, contact number, category, userID)

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)

Rating Outputs:

- Displaying overview of services (title, area, category, offer/request)
- Displaying service details (title, price, description, offer/request, area, contact number, category, email)
- Display User Account information (username, email, contact number, userID)
- Display user's own posted service (title, price, description, offer/request, area, contact number, category, email)

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

Rating Inquiries:

- Filtering the services based on user's criteria

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

Rating Logical Files:

- Service's information
- User's account information

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)

Rating Interfaces:

- Firebase authentication
- Firebase real-time database

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)

Summary of above analysis:

Element	Complexity	Detail
Inputs	Low	User Registration
	Low	User Login
	Low	Reset password
	Low	Delete Service Posts
	Low	Modifying user's account information
	Average	Request/Offer Services
Logical Files	Low	Service's information
	Low	User's account information
Outputs	Low	Display overview of Services
	Low	Display Service Details
	Low	Display User Account information
	Average	Display User's own posted service
Inquiries	Low	Filtering the services based on the user's criteria
Interfaces	Low	Firebase authentication
	Average	Firebase real-time database

Calculation of Unadjusted Function Points:

Characteristic	Low		Medium		High	
Inputs	5	× 3	1	× 4	0	× 6
Outputs	3	× 4	1	× 5	0	× 7
Inquiries	1	× 3	0	× 4	0	× 6
Logical Files	2	× 7	0	× 10	0	× 15
Interfaces	1	× 7	1	× 10	0	× 15
Unadjusted FP	51		19		0	
Total=L+M+H	70					

5.1.2 Adjusted Function Points

Influence Factors	Score	Detail
Data Communications	5	Application is more than a front-end, and supports more than one type of teleprocessing communications protocol.
Distributed Functions	4	Distributed processing and data transfer are online and in both directions.
Performance	3	Response time or throughput is critical during all business hours. No special design for CPU utilisation was required. Processing deadline requirements with interfacing systems are constraining.
Heavily used	2	Some security or timing considerations are included.
Transaction rate	3	Daily peak transaction period is anticipated.
On-line data entry	5	More than 30% of transactions are interactive data entry
End-user efficiency	2	Four to five of the efficiency designs are included
On-line data update	3	Online update of major internal logical files is included.
Complex processing	1	Any one of the complex components
Reusability	4	The application was specifically packaged and/or documented to ease re-use, and the application is customised by the user at source code level.
Installation Ease	1	No special considerations were stated by the user *but* special setup is required for installation.
Operational Ease	1	Effective start-up, back-up, and recovery processes were provided, but no operator intervention is required (count as two items).
Multiple sites	0	User requirements do not require considering the needs of more than one user/installation site.
Facilitate change	3	Flexible query and report facility is provided that can handle complex requests, for example, *and/or* logic combinations on one or more internal logical files (count as three items).
Total score	37	
Influence Multiplier = Total score × 0.01 + 0.65 = 37 × 0.01 + 0.65 = 1.02		
Adjusted FP = Unadjusted FP × Influence Multiplier = 70× 1.02 = 71.4		

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

5.1.3 Lines of Code

According to Capers Jones statistics, each Function Point requires 53 lines of code if the application is implemented using Java.
Therefore, we have: **Lines of Code** = 71.4 FP × 53 LOC/FP = **3784 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.

- Working days include 5 days in a week.
- Effort = Size / Production Rate = (3784 LOC) / (39 LOC/PD)[1] = 97 PD
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (97) $^{1/3}$ = 13.79 Days
- Initial schedule = 13.79 Days / 5 days a week = 2.76 Weeks
- Team size = 97 PD / 13.79 D = 7.03 P = 7 Persons
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 97 PD × 8 hours = 776 PH

5.2.1 Distribution of Effort

1990's Industry Data	Work Package	Distribution	Estimates
Preliminary Design 18 %	Project Plan	9%	69.84
	Requirement Specification	9%	69.84
Detailed Design 25 %	User Interface	7%	54.32
	Technical Architecture	11%	85.36
	Data Modelling	7%	54.32
Code & Unit Testing 26 %	Code & Unit testing	21%	162.96
	Online Documentation	5%	38.8
Integration & Test 31 %	Integration & Quality Assurance	31%	240.56
	Extrapolated total effort		776
	2% for project management		15.52
	3% for contingency		23.28
	Total effort		814.8

These duration estimates are based on the assumption 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 - Dell Computer	Total $8200.00
12th Gen Intel® Core™ i5-12400	
8 GB, DDR4	
256 GB SSD + 1 TB HDD	
2 - Samsung Galaxy A53 5G	
6 GB	
Exynos 1280	
256GB	

Software:

GNU, Apache, or Other Free Licence-based Software:

Firebase	$15 000.00

Software Licence Provided by Third Party:

Google Docs	$0.00
Zoom	$0.00
Visual Paradigm	$0.00

Other Resources:

Staff:

6 employees with 776 working hours with $4/hr ($3104/project)	$18,624.00
Project Manager with 776 working hours with $40/hr ($31040/project)	$31,040.00

Miscellaneous:

Transportation	$100.00

Total:
$79, 864.00

Think2's hardware and software responsibilities relate only to our own development needs to accomplish the project we have been asked to complete, and which has been described in the introduction section of this document. Think2 will also demonstrate the completed product.

6 Product Checklist

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

Project Deliverable	Estimated Deadline
Project Plan	6 October 2022
Requirements Specification	15 September 2022
Design Document	20 October 2022
Module/System Test Plan	20 October 2022
System Release (Demo)	20 October 2022

7 Best Practice Checklist

Practice	9
Document what we do; all documentation must be in a standardised 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: • Minimise interfaces between modules, procedures and data. • Minimise 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 designs, 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 the general risk management, the following risks have been identified for the HelpMe project:

Undermined motivation
Impact Severity: High
Probability: 25%
Impacts: We may not be motivated to complete all deliverables on time.
Risk Reduction: Good team-building, clear and equal distribution of tasks

More changes to requirements than anticipated
Impact Severity: High
Probability: 25%
Impacts: Depending on the stage at which changes occur, could range from needing to update the requirements documentation to needing to do a complete redesign.
Risk Reduction: Be rigorous in eliciting requirements.

System size underestimated
Impact Severity: Moderate
Probability: 30%
Impacts: More work will need to be spent on design and coding; could negatively impact schedule.
Risk Reduction: Update estimates often as the project progresses.

Staff leaving before project complete
Impact Severity: Low
Probability: 1%
Impacts: There would be more work for remaining employees, and any specialised skills or knowledge would be lost.
Risk Reduction: Offer benefits and incentives to staff.

Problems coordinating within group
Impact Severity: Moderate
Probability: 40%
Impacts: Members may be unaware of what is expected of them; managers may not be able to measure progress; portions of projects not completed.
Risk Reduction: Follow communication plans as documented in section 2.3

Database limits
Impact Severity: Moderate
Probability: 25%
Impacts: Depending on the user traffic and number of requests, the application may give delayed responses or no response at all.
Risk Reduction: Redundant and reliable database infrastructure

9 Quality Assurance

The project will achieve quality assurance by following the standard set by the company. The specific procedures and details shall be provided in the Quality Plan.

Specific test procedures and details shall be provided in the Module/System Test Plan.

In addition, the HelpMe application shall make use of two testing methodologies:

- **Unit Testing** involves testing system components individually.
- **In-Place Testing** involves testing of the whole system as a unit.

Furthermore, these methodologies will be used to test two important aspects of the application:

- **System Function** will be tested to ensure that software flaws are eliminated, and
- **Algorithmic Function** will be tested to ensure that heuristic aspects of the project perform realistically to provide value to the users.

We will also check for the following standard metrics:

- **Fan in/Fan-out**

  Fan-in is a measure of the number of functions or methods that call some other function or method (say X). Fan-out is the number of functions that are called by function X. A high value for fan-in means that X is tightly coupled to the rest of the design and changes to X will have extensive knock-on effects. A high value for fan-out suggests that the overall complexity of X may be high because of the complexity of the control logic needed to coordinate the called components.
- **Length of code**

  This is a measure of the size of a program. Generally, the larger the size of the code of a component, the more complex and error-prone that component is likely to be. Length of code has been shown to be one of the most reliable metrics for predicting error-proneness in components.
- **Cyclomatic complexity**

  This is a measure of the control complexity of a program. This control complexity may be related to program understandability.
- **Mean Time To Failure**

  The average time between two system failures should be sufficiently large.
- **Depth of conditional nesting**

  This is a measure of the depth of nesting of if-statements in a program. Deeply nested if statements are hard to understand and are potentially error-prone.
- **Depth of inheritance tree**

  This represents the number of discrete levels in the inheritance tree where subclasses inherit attributes and operations (methods) from superclasses. The deeper the inheritance

tree, the more complex the design. Many different object classes may have to be understood to understand the object classes at the leaves of the tree.

Our methodology makes broad use of realistic test cases. Detailed test data is an important part of the final project delivery.

10 Monitoring & Control

Many procedures are required in order to be able to successfully monitor the progress of a software project. Some of the most important are:

Quantitative measurement of resource consumption: Estimates of the application's resource requirements, primarily in terms of human resources, can provide a quantitative measurement of project progress when compared to progress in terms of project milestones. The percentage estimates of each milestone's resource requirements provided in this document allow for easy progress tracking.

Identification of major project risks: Early identification of major risks to the project allows for placement of preventative measures before problems can develop. Major risks have been identified in the Risk Management section of this document, along with the measures being taken to avoid them.

Risk Response Planning: The analysed risks will be categorised into red, yellow and green zones according to criticality. The framework of avoid, mitigate, accept and transfer will be used to address the risks. Risks will be monitored continuously, and the other team members will be notified of important changes to the risk assessment.

Regular reviews of project progress: Throughout the duration of the project, the team shall meet weekly to review the progress of all project tasks, including management, planning, analysis, development, and testing.

Timeline Planning and task decomposition: This document outlines an estimated timeline for the project. A reasonably accurate timeline can be assembled by hierarchically decomposing tasks into measurable subcomponents and estimating requirements for each. At the same time, this decomposition can assist in task assignment and balancing. Throughout the implementation phase, these subcomponents can allow for fine-grained measurement of progress. Project subcomponents and timeline estimates are included in the Estimates and Work Breakdown Structure sections of this document.