Project Plan for “Go where GaiGai” web application

[Project Plan]
Version 1.0

By: Eugene Lim, Jasper Yeo, Chang Tze Chuan, Isaac Soh, Huang RuiMin, Yap Wee Kiat

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	29 September 2022	Eugene Lim	First version

Table of Contents

1 Introduction

1.1 Project Overview

"Go Where GaiGai" is a website that seeks to make it quick and easy for people to find popular dining and entertainment destinations near them. It uses the user's current location and several user preferences to search for and display the destinations on a map. From the map, the user can then view information about these locations and choose to add them into a planner to share with their friends.

1.2 Project Description and Scope

"Go Where GaiGai" is a software project undertaken by Group ONE. "Go Where GaiGai" aims to make searching and planning for food and leisure experiences quick and simple. The website allows users to search for food and entertainment destinations, by accepting search preferences such as location and category. The website will incorporate the use of an intelligent algorithm that will select and suggest the top five locations based on user reviews, ratings and proximity of the location relative to the user's current location. The user is then able to view basic information about the suggested locations, and add them into a planner for sharing. The planning module allows the user to plan out activities and import/export them for sharing. The back-end of this system includes a database containing the geographic information used by the search algorithm and map. The front-end user interface will poll the database at regular intervals to ensure the latest information is used to service the website.

Features of the system include:

1. Interactive Map
2. Interactive Location Map Pins
3. Location Search Algorithm
4. Activity Planner
5. Activity Plan Viewer
6. Importing/Exporting Activity Plan

Other features can be incorporated into the software following client, customer and stakeholder feedback.

2 Project Organization

2.1 Team Structure

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

- Project Manager: Eugene Lim Zhi Jie
- Lead Developer: Yeo Kai Liang Jasper
- Front-end Developer: Chang Tze Chuan
- Back-end Developer: Isaac Soh Wei Yang
- QA Manager: Huang RuiMin
- QA Engineer: Yap Wee Kiat
- Release Manager: Eugene Lim Zhi Jie

2.2 Roles and Responsibilities

Project Manager: Eugene Lim Zhi Jie

- Oversees project progress
- Approves and executes project plan
- Assigns tasks and reports status of project to team members
- Represents the team to upper management/client
- Manages and motivates team members

Lead Developer: Yeo Kai Liang Jasper

- Collects information from client interviews
- Develops concepts of the system for designers
- Creates requirement specification document
- Designs logical system based on requirements

Front-end Developer: Chang Tze Chuan

- Translates logical design into detailed design
- Creates detailed design document
- Implements product based on detailed design document
- Designs User Interface

Back-end Developer: Issac Soh Wei Yang

- Ensures stability and response time of the system meet the requirements
- Integration of coded modules into functioning system

Quality Assurance Manager: Huang RuiMin

- Ensures acceptable software quality
- Designs testing strategies
- Creates and manages test plan

Quality Assurance Engineer: Yap Wee Kiat

- Executes test procedures
- Verify software requirements

Release Manager: Eugene Lim Zhi Jie

- Creates user manual
- Creates build baselines and integrate changes for delivery

2.3 Team Communication

Communication channels of Group ONE include the following:

- Weekly meetings are held on Mondays at 8:30 am.
- Group announcements and updates are disseminated through Project Manager's email: euge0027@e.ntu.edu.sg
- Online Zoom discussions are held as necessary between weekly meetings.
- Weekly/Online meeting minutes will be made available on Team Wiki.
- Split up into subgroups as necessary, in order to work more cooperatively on specific problems.
- File sharing using Google Drive and Github.

3 Process Definition

3.1 Lifecycle Model

During the development of “Go Where GaiGai”, the project will adopt a plan-driven Incremental Development Model. Due to the short timeline of this project, the Incremental Development Model will allow the development team to gain feedback from potential users quickly, and allow them to implement changes and bug fixes quickly. The builds from each incremental cycle can then be used by the quality assurance team to perform unit and integration tests concurrently with the assistance of the Release Manager. The initial cycles will see to it that major features of “Go Where GaiGai” are implemented with the highest priority, with subsequent builds ensuring all other system requirements are met.

Other software development methodologies such as Waterfall model, Spiral Model and V-Model have been taken into consideration. Reasons for avoiding these development models include the dynamic nature of this project. The Waterfall model will not allow the development team to backtrack to previous phases in the project plan, incurring increased costs in time and effort should previous phases be executed inaccurately. Based on the Waterfall model, the Spiral model faces the same difficulty, where the timeline of the project will not allow the team to split phases of the development cycle into individual iterative cycles. The V-model allows for different sub-groups to progress in parallel. However, the V-model does not allow the team to transition between the major phases of software development if any sub-task of the previous phases is incomplete.

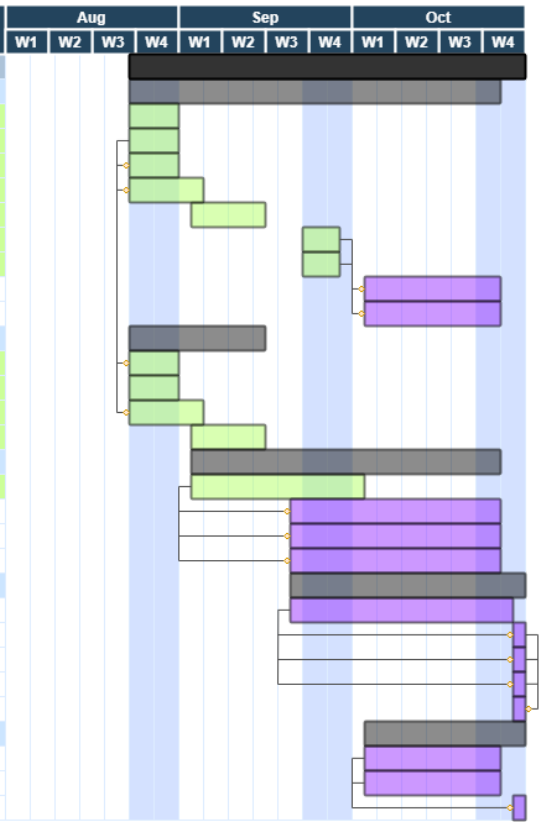
Group ONE intends to deliver the first iteration of the system prototype on the Prototype Delivery Date as indicated in the Project Estimates section of this document. After future client interaction and approval from management, further iterations will be released as necessary.

Due to the intelligent nature of the search algorithm of “Go Where GaiGai”, tweaking and fine-tuning of the search algorithm with respect to client feedback will be performed throughout the course of the software timeline. This is done to ensure the algorithm behaves in an accurate and logical manner, providing a more effective search experience for users on the website.

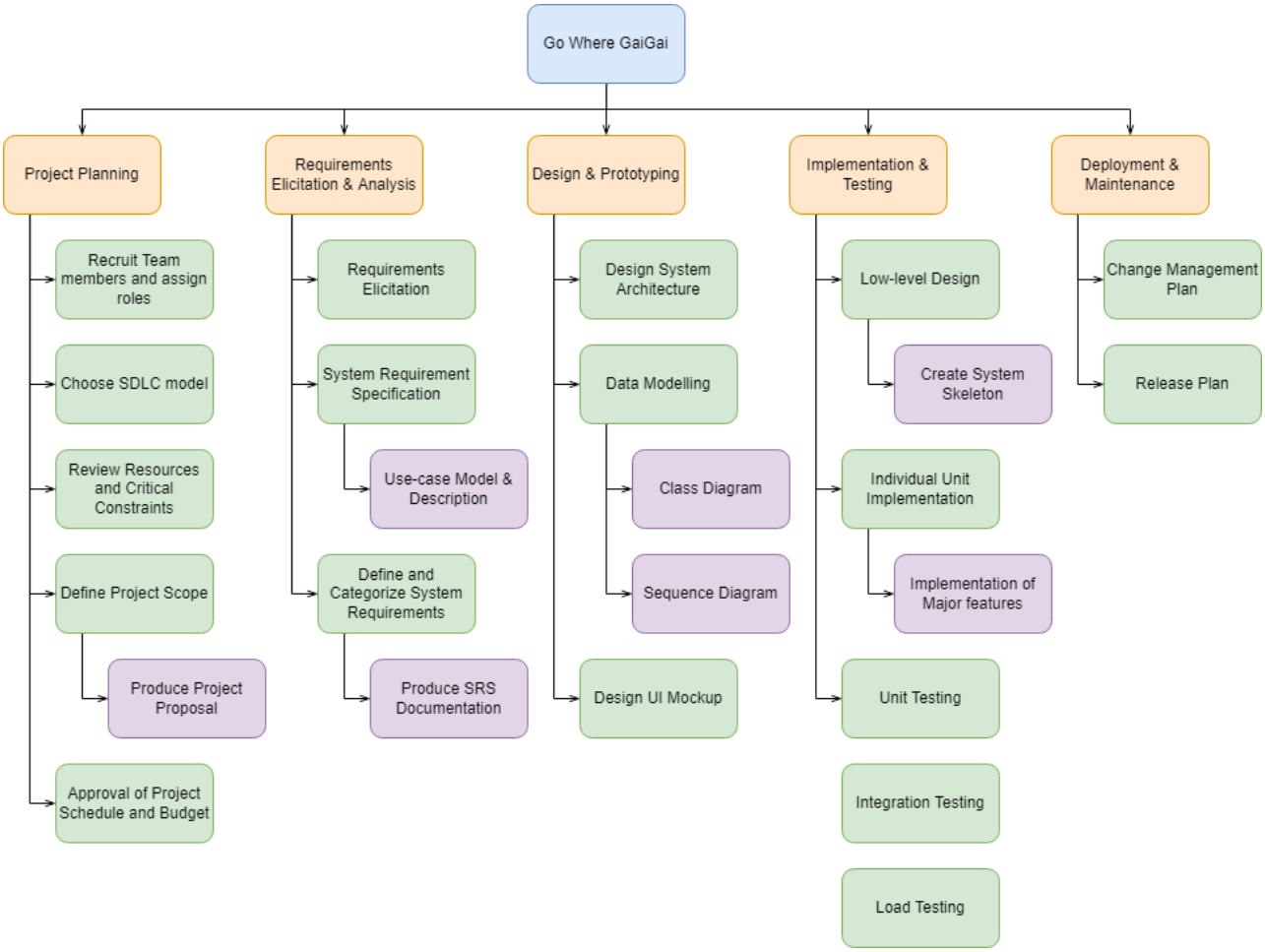
4 Schedule

4.1 Activity Dependencies and Schedule

	Task	Start	Due	%
	Complete project execution	22/8	31/10	48%
	Project Planning			78%
1	Define Project Scope	22/8	29/8	100%
2	Produce Project Proposal	22/8	29/8	100%
3	Produce Task Backlog	22/8	29/8	100%
4	Setup Team Wiki	22/8	5/9	100%
5	Produce Quality Management Plan	5/9	12/9	100%
6	Produce Project Plan	19/9	26/9	100%
7	Produce Risk Management Plan	19/9	26/9	100%
8	Produce Configuration Management Plan	3/10	17/10	0%
9	Produce Change Management Plan	3/10	17/10	0%
	Design			100%
10	Produce Use-case model	22/8	29/8	100%
11	Produce Initial UI Mockups	22/8	29/8	100%
12	Design System Architecture	22/8	5/9	100%
13	Complete System Requirement Specifications	5/9	12/9	100%
	Implementation			25%
14	Create Prototype	5/9	3/10	100%
15	Implement Main Page	12/9	17/10	0%
16	Implement Search Algorithm	12/9	17/10	0%
17	Implement Planner	12/9	17/10	0%
	Testing			0%
18	Produce Test Plan	17/10	24/10	0%
19	Unit Test	24/10	31/10	0%
20	Integration Test	24/10	31/10	0%
21	Load Test	24/10	31/10	0%
22	Produce Requirements Test Coverage Report	24/10	31/10	0%
	Deployment			0%
23	Produce Release Plan	3/10	17/10	0%
24	Produce Software Maintainability Report	3/10	17/10	0%
25	Deployment of Application	27/10	31/10	0%



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:

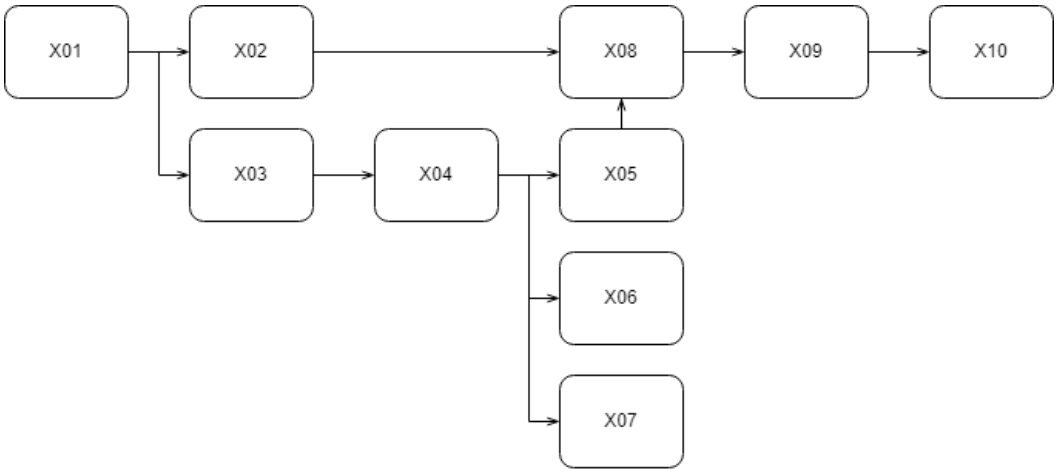
1. Project Proposal
2. Project Plan
3. Requirements Elicitation
4. Requirements Analysis and Documentation
5. System Architecture Design
6. Data Modeling
7. Prototype and User Interface Design
8. Coding and Unit Testing
9. Integration and Quality Assurance
10. Product Deployment

4.4 Activity Dependencies

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

Work Package #	Work Package Description	Duration (days)	Dependencies
X01	Project Proposal	7	--
X02	Project Plan	7	X01
X03	Requirements Elicitation	4	X01
X04	Requirements Analysis and Documentation	7	X03
X05	System Architecture Design	4	X04
X06	Data Modeling	4	X04
X07	Prototype and User Interface Design	4	X04
X08	Coding and Unit Testing	35	X05
X09	Integration and Quality Assurance	7	X08
X10	Product Deployment	4	X09

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



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	Go Where GaiGai
Work Package	X01 - Project Proposal (1 of 10)
Assigned To	**Eugene Lim**
Effort	7 PD
Start Date	Monday, 22 August 2022
Purpose	To create an introductory overview of the project, to be submitted for approval to management.
Inputs	None
Activities	This work package includes the problem statement, objective of the project. The proposal also describes the technical approach adopted by the team in creating the website. A set of proposed project deliverables as well as budget and communication plans will be included in the Project Proposal. From these requirements, a team of members will be recruited based on their possessed qualifications.
Outputs	A written document of the Project Proposal.

Project	Go Where GaiGai
Work Package	X02 - Project Plan (2 of 10)
Assigned To	**Eugene Lim**
Effort	7 PD
Start Date	Monday, 5 September 2022
Purpose	To allow planning of the project schedule to work within various constraints and adhere to delivery timelines as close as possible.
Inputs	Time estimations on delivery of prototype and unit implementation required from the Development team based on experience.

Activities	This work package will include descriptions of software estimation techniques, product and best practice checklists, so that the Project Manager can perform resource and task allocation. This work package will form the baseline on how the project is monitored and controlled, to ensure the team delivers the requirements specified in the delivery timelines.
Outputs	A written document of the Project Plan.

Project	Go Where GaiGai
Work Package	X03 - Requirements Elicitation (3 of 10)
Assigned To	**Jasper Yeo**, Chang Tze Chuan, Eugene Lim
Effort	4 PD
Start Date	Thursday, 25 August
Purpose	To establish a common understanding between the client and the software project team of the requirements to be addressed in the project.
Inputs	Client's requirements and potential customer's feedback
Activities	The Lead Developer will hold a meeting with the client to determine the requirements of the system. He will also interview potential users to produce a set of requirements that will improve the quality of user experience and ease of access on the website.
Outputs	A written document of the Requirement Specification.

Project	Go Where GaiGai
Work Package	X04 - Requirements Analysis and Documentation (4 of 10)
Assigned To	**Jasper Yeo**, Chang Tze Chuan, Eugene Lim
Effort	7 PD
Start Date	Thursday, 1 September 2022
Purpose	To identify the main purpose and direction of the system, critical constraints during the development cycle and major features required of the developed product.
Inputs	Requirement Specifications from previous work package (X03).
Activities	High level functional and nonfunctional requirements must be captured in this work package. The requirements must then be made atomic, verifiable and

	testable. The Use-case Model and descriptions will be produced in this documentation such that each use case is traceable to their respective design documentation, source code and test cases.
Outputs	A written document of System Requirement Specification (SRS), and Use-case Model and Description.

Project	Go Where GaiGai
Work Package	X05 - System Architecture Design (5 of 10)
Assigned To	**Isaac Soh**, Jasper Yeo, Chang Tze Chuan
Effort	4 PD
Start Date	Thursday, 1 September 2022
Purpose	To define how components of the software will interact and select appropriate packages and dependencies to be used in this project.
Inputs	Technology Considerations from the Project Proposal and System Requirement Specification document.
Activities	High level design entails defining the architecture of the software system and identifying the various components and how they interact with each other. The platforms and tools used in this project will be selected in this work package based on the technological considerations performed in the Project Proposal. The Lead Developer will also produce a System Architecture Diagram which visualizes the relationships between the different components and levels of abstraction of the software.
Outputs	A written System Architecture Diagram.

Project	Go Where GaiGai
Work Package	X06 - Data Modeling (6 of 10)
Assigned To	**Jasper Yeo**, Chang Tze Chuan, Isaac Soh
Effort	4 PD
Start Date	Thursday, 1 September 2022
Purpose	To visualize and establish a common understanding between the software project team of how the system works and is designed.
Inputs	System Requirement Specification document.

Activities	The development team will produce this work package while working on System Architecture Design (Work Package X05) simultaneously. A UML Class diagram and Sequence diagram will be made for formal documentation in the software development process.
Outputs	Written UML Class diagram and Sequence diagram.

Project	Go Where GaiGai
Work Package	X07 - Prototype and User Interface Design (7 of 10)
Assigned To	**Jasper Yeo**, Chang Tze Chuan, Isaac Soh, Yap Wee Kiat
Effort	4 PD
Start Date	Monday, 5 September 2022
Purpose	UI Mockup to serve as a guiding representation of the final product. The Prototype will be used to gather client feedback and ensure major features are implemented with minimal bugs.
Inputs	User feedback, and Data Modeling documents.
Activities	The development team with the assistance of the Project Manager will create a UI mockup. A meeting with the full software project team and client will ensure all parties are aware of the design of the website. After approval, prototyping of "Go Where GaiGai" will begin immediately.
Outputs	Graphical UI Mockup and working prototype.

Project	Go Where GaiGai
Work Package	X08 - Coding and Unit Testing (8 of 10)
Assigned To	**Jasper Yeo**, **Huang RuiMin**, Change Tze Chuan, Isaac Soh, Yap Wee Kiat
Effort	35 PD
Start Date	Monday, 12 September 2022
Purpose	To fulfill system requirements and begin Unit Tests against individual Unit implementations/functions.
Inputs	Working prototype for coding, and System Requirements Specification for creating test cases.
Activities	This work package covers the majority of the implementation phase. The iterative development cycle will ensure the website reaches a state at which all

	system requirements are fulfilled. The QA Manager will also produce the System Test Plan for the QA Engineer to perform Unit Tests in this work package and ensure test cases are verifiable and testable.
Outputs	Source code and Header files. A written System Test Plan will also be produced.

Project	Go Where GaiGai
Work Package	X09 - Integration and Quality Assurance (9 of 10)
Assigned To	**Huang RuiMin**, Yap Wee Kiat, Jasper Yeo, Chang Tze Chuan, Issac Soh, Eugene Lim
Effort	7 PD
Start Date	Monday, 17 October 2022
Purpose	To integrate each individual unit code into a main system and ensure the data flow through this system is correct. This work package also identifies and allows the team to solve logical and syntactical errors during testing.
Inputs	System Test Plan and working code from previous packages (X08, X07).
Activities	The development team and Release Manager will integrate the system components by simulation potential users may interact with the website. The Back-end Developer will use measurable metrics to conduct system performance and integrity tests. After the system functions as per requirements, the QA Team will complete integration testing and load testing with the assistance of the development team. Heuristics assessment plays an important role in this work package, as intelligence components will define eventual system success.
Outputs	Source code, Test files and a written Test Coverage Report.

Project	Go Where GaiGai
Work Package	X10 - Product Deployment (10 of 10)
Assigned To	**Eugene Lim**, Jasper Yeo, Chang Tze Chuan, Issac Soh, Huang RuiMin, Yap Wee Kiat
Effort	4 PD
Start Date	Thursday, 27 October 2022
Purpose	To deploy the system and provide clients with user manuals and update/release

	plans.
Inputs	Software Maintainability Report and Release Plan.
Activities	Release Manager to ensure all documentation is submitted properly, and the working system is delivered successfully to the client, along with user manuals for training.
Outputs	Code repository and all development documentation archived properly.

5 Project Estimates

5.1 Code Size Estimation using Function Points

In this section, project estimation will be conducted using Function Point analysis. The functional size of the product will be measured in terms of function points, which is a standard unit of measurement to measure the complexity of software application. By using a measurable unit such as function points, the software project development along with testing and maintenance can be consistently quantified and used to provide project estimates to clients, customers and stakeholders.

The calculation of unadjusted function points is based on five primary elements:

1. Number of Inputs
2. Number of Outputs
3. Number of Inquiries
4. Number of Logical Files
5. Number of External Interfaces.

The unadjusted function point values of each of the five elements will be calculated by assigning a complexity factor. After which, several other drivers such as data communications, transaction rates etc. will be taken into account to derive an **influence multiplier**. Once the unadjusted function point value is multiplied with the influence multiplier, the **adjusted function point** value can be found. Lastly, the adjusted function point value can be converted into estimated lines of code for project estimation.

5.1.1 Unadjusted Function Points

"Go Where GaiGai" supports the following proposed functions:

7. Interactive Map
8. Interactive Location Map Pins
9. Location Search Algorithm
10. Activity Planner
11. Activity Plan Viewer
12. Importing/Exporting Activity Plan

The measure of unadjusted function points is based on five primary component elements of these functions: Number of Inputs, Number of Outputs, Number of Inquiries, Number of Logical Files, and Number of External Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity. The detailed evaluation of the complexity is as follows:

Rating Inputs:

- Location Search Algorithm: User inputs search preferences (categories, location, postal code) - **1 input (low)**
- Interactive Location Map Pins: User can add location to the Activity Planner - **1 input (low)**
- Activity Planner: Users are able to insert, delete, update locations in the Activity Planner - **3 input (High)**
- Importing/Exporting Activity Plan: Users are able to import a plan from their computer - **1 input (Medium)**

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

Rating Outputs:

- Interactive Map: Interactive Map is displayed to the user - **1 output (Low)**
- Activity Plan Viewer: System to display plans to the user - **1 output (Low)**
- Interactive Map: Details of location are displayed when the user clicks on pins (image, name, ratings, category, address, reviews of location) - **1 output (Medium)**

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

Rating Inquiries:

- Location Search Algorithm: System displays and highlights drop points of the search result - **1 inquiry (Medium)**
- Interactive Map: User able to zoom in and out of map, and reset map - **1 inquiry (Low)**
- Importing/Exporting Activity Plan: Users able to export a plan for download - **1 inquiry (Low)**
- Importing/Exporting Activity Plan: System checks if imported plan is valid, and returns message - **1 inquiry (Low)**

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

Rating Logical Files:

- Location Search Algorithm: Location data - **1 input (High)**

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

Rating Interfaces:

- Location Search Algorithm: System queries the database for results based on user search - **1 interface (Low)**

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

6 or More RET	Medium (10)	High (15)	High (15)

Summary of above analysis:

Element	**Complexity**	**Detail**
Inputs	Low	User inputs search preferences
	Low	User can add location to the Activity Planner
	High	Users are able to insert, delete, update locations in the Activity Planner
	Medium	Users are able to import a plan from their computer
Outputs	Low	Interactive Map is displayed to the user
	Low	System to display plans to the user
	Medium	Details of location are displayed when the user clicks on pins
Inquiries	Medium	System displays and highlights drop points of the search result
	Low	User able to zoom in and out of map, and reset map
	Low	Users able to export a plan for download
	Low	System checks if imported plan is valid, and returns message
Logical Files	High	Location data
Interfaces	Low	System queries the database for results based on user search

Calculation of Unadjusted Function Points:

Characteristic	**Low**		**Medium**		**High**	
Inputs	2	× 3	1	× 4	3	× 6
Outputs	2	× 4	1	× 5	0	× 7
Inquiries	3	× 3	1	× 4	0	× 6
Logical Files	0	× 7	0	× 10	1	× 15
Interfaces	1	× 5	0	× 7	0	× 10
Unadjusted FP	28		13		33	
Total=L+M+H	74					

5.1.2 Adjusted Function Points

Influence Factors	**Score (0-5)**	**Detail**
Data Communications	4	The application refreshes locations and user reviews/ratings at regular intervals to be displayed on the front-end. Hence data communication is significantly involved.
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 utilization was required. Processing deadline requirements with interfacing systems are constraining.
Heavily used	2	The application is expected to experience moderate to low loads.
Transaction rate	3	Daily peak transaction period is anticipated.
On-line data entry	5	More than 50% of transactions are interactive data entry.
End-user efficiency	1	End-user efficiency is not a major consideration.
On-line data update	3	Online update of major internal logical files is included.
Complex processing	1	Complex processing is essential when user preferences are passed into the search function.
Reusability	4	The application was specifically packaged and/or documented to ease re-use, and the application is customized by the user at source code level.
Installation Ease	2	No special considerations are required for installation as the system is fully hosted on a web server.
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	36	
Influence Multiplier = Total score × 0.01 + 0.65 = 36 × 0.01 + 0.65 = 1.01		

Adjusted FP = Unadjusted FP × Influence Multiplier = 88 × 1.01 = 88.88

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 Javascript, the language framework for ReactJS.

Therefore, we have: **Lines of Code** = 88.88 FP × 53 LOC/FP = **4711 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 = (4711 LOC) / (39 LOC/PD)[1] = 121 PD
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (121) $^{1/3}$ = 14.8 Days
- Initial schedule = 14.8 Days / 5 days a week = 2.96 Weeks
- Team size = 121 PD / 14.8 D = 8.17 P = 8 Persons
- Working hours include 8 hours in a working day.
- Total person-hours (PH) = 121 PD × 8 hours = 968 PH

5.2.1 Distribution of Effort

1990's Industry Data		**Work Package**	**Distribution**	**Estimates**
Preliminary Design 18 %		Project Plan	9%	87.12
		Requirement Specification	9%	87.12
Detailed Design 25 %		User Interface	7%	67.76
		Technical Architecture	11%	106.48
		Data Modeling	7%	67.76
Code & Unit Testing 26 %		Code & Unit testing	20%	193.6
		Online Documentation	6%	58.08
Integration & Test 31 %		Integration & Quality Assurance	31%	300.08
		Extrapolated total effort		968
		2% for project management		19.36
		3% for contingency		29.04
		Total effort		1016.4

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	**Cost**
8 x Lenovo Desktop	$2,500.00
Single Core	
1 GHz	
4 GB Ram	

Software:

GNU, Apache or other Free License software	**Cost**
Apache Web Server	$0.00
Perl	$0.00

Third Party Licenses	**Cost**
Microsoft Office 2000	$0.00
Microsoft Project 2000	$0.00
Projects ESTIMATE Professional	$0.00

Others:

Staff	**Cost**
8 Employees with 968 working hours at $18.00/hour	$17,424.00

Stationery	**Cost**
Paper, photocopying and other miscellaneous costs	$50.00

Total development costs amount to **$19974.00**. The customer will supply the required hardware and software necessary to run the back-end server and deploy the web server for "Go Where GaiGai". Group ONE is not responsible for supplying said systems. Group ONE's hardware and software responsibilities relate only to their development needs to accomplish the project, and which has been described in the introduction section of this document. Group ONE 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	Oct 4th, 2001
Requirements Specification	Oct 18th , 2001
Design Document	Oct 25th, 2001
Module/System Test Plan	Nov 8th , 2001
System Release (Demo)	Nov 15th, 2001

7 Best Practice Checklist

Practice
All models adopted by the team as well as actions taken will be documented formally and in a standardized format. The standardized format warrants: - Consistent Fonts, Font colors and sizes for headers and bodies. - Consistent line spacing and indentation. - Easy to read and understand with minimal need to refer to the project's data dictionary.
System Requirements/Specifications documented must follow industry standards in order to ensure that the needs of clients and stakeholders are met. - Functional requirements must be atomic, verifiable and testable. - Nonfunctional requirements must be verifiable and testable. In addition, all requirements must be consistent, precise and unambiguous. Should any requirements have any critical constraints, the constraints must be indicated clearly on each piece of documentation.
It is important to ensure that the program does not become too complex, or contain too much code. A larger code base inherently contains more bugs and data flow problems. To meet the customer's requirements, the team must: - Minimize interfaces between software modules, procedures and data. - Minimize interfaces between people, otherwise an exponential increase in communication cost will occur. - Prioritize working product functions and structure over aesthetic design choices.
Require Visibility. There must be transparency behind the processes and the work produced by each member of the project team. This will allow Group ONE to measure progress accurately throughout the development cycle of the product and be able to take management actions as appropriate. This includes: Proper communication between the Project Manager and the Lead Developer and QA Manager, as well as the Lead Developer and QA Manager and their respective team members. Source and test code are also to be made available to review at each revision before changes are pushed into the team's repository.
Plan for continuous change. The team must: - For all user manual designs, test/source code have version numbers, dates, revision history comments and change marks to indicate the changes. - New revisions should be approved before being made and checked for quality and compliance before being pushed into the team's common repository. - Use a configuration management system and create systematic procedures when

applying changes in code or documentation.
The team 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 peer code reviews between team members.
The following points must be considered when conducting testing. - The test cases must be as exhaustive as possible in ensuring complete test coverage in order to simulate realistic scenarios. - Every requirement must be tested by a test case. - Test cases must cover error handling scenarios. - Boundary conditions must be correctly identified and test cases for such scenarios must be crafted.

8 Risk Management

Besides the general risk management, the following risks have been identified for the software project:

Location information database overloads
Impact Severity: Medium
Probability: 25%
Impacts: The interactive map will be functioning based on outdated information. Users might not receive accurate search results, based on said outdated information. Otherwise, response time of the system will be delayed greatly.
Risk Reduction: Investment in a highly redundant, reliable and elastic database infrastructure.

Interfacing between front and back-end malfunctions
Impact Severity: High
Probability: 10%
Impacts: The interactive map will not be able to display location drop points after the system searches for locations. Website will not be operational beyond a basic map function.
Risk Reduction: Ensure code is peer reviewed and tested extensively for failure of front and back-end interfacing.

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. Make sure customers are aware of potential repercussions of requirement changes.

Specification Delays
Impact Severity: High
Probability: 15%
Impacts: Delays in finalizing the specification will push the schedule for all following stages of the project.
Risk Reduction: Monitor progress of specification carefully.

Underestimating System size
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 the project completes
Impact Severity: Extreme
Probability: 5%
Impacts: There would be more work for remaining employees, and any specialized 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

Customer cancels project
Impact Severity: Super-Extreme!
Probability: 1%
Impacts: All work will have been wasted.
Risk Reduction: Keep in close contact with customers. Ensure that they have some market research indicating a demand for this product.

9 Quality Assurance

The project will achieve a level of 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, “Go Where GaiGai” shall make use of the following testing methodologies:

- **Unit Testing**
  Each individual unit of code implemented in the creation of “Go Where GaiGai” will be tested in isolation. By separating units before testing, the Quality Assurance Team can use white box testing as an efficient and inexpensive method to ensure all functions are working as intended. Proper results are determined by designing test cases with the appropriate input boundaries before testing for all valid and/or invalid outputs from each unit. The usage of white box testing on unit tests guarantees complete test coverage per unit of code.
- **Integration Testing**
  Although each unit is fully tested in Unit Testing, Integration Testing is required to ensure different groups of units interact together properly. Due to the iterative development cycle adopted in this software project, all software modules might see functional and nonfunctional changes due to the changing requirements derived from client feedback. Coupled with the fact that software modules are written by different team members, each unit might have different implementation logic. Hence, Integration Testing is conducted to ensure that all modules interact with each other logically and accurately and expose defects when data flows from one unit to another.
- **Performance Testing**
  Performance Testing will be conducted after implementation of “Go Where GaiGai” to certify that the final product is ready for deployment, by checking nonfunctional requirements that indicate how well the system performs under realistic scenarios. Performance Testing is broken down into numerous sub-tests. However, the tests used include load, stress, endurance and scalability testing. Load tests will measure the system’s performance in terms of response time when concurrent users and system transactions increase. Stress testing is an extension of load testing wherein a system is fed abnormally high concurrent users and system transactions to measure the working limits of the system and how well the system can be recovered from failure. Endurance testing will monitor how well the system manages its resources after operating for extended lengths of time. Lastly, Scalability testing will indicate if the system is able to handle increasing workloads both on the front and back-end effectively. This helps the developers identify any need to expand databases or refactor front-end code.

Furthermore, these methodologies will be used to test two important aspects of “Go Where GaiGai”:

- **System Function** will be tested to ensure that software flaws are eliminated, and
- **Algorithmic Search Function** will be tested to ensure that heuristic aspects of the project (such as user preferences) perform realistically to provide users with the best search experience.

Group ONE’s methodology makes broad use of realistic test cases. Detailed test data is an important part of the final project delivery and will be delivered as a Test Coverage Report.

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:

Identification of major project risks: An early identification of the major risks that might develop throughout the project's development life cycle gives the team time to plan for preventive measures. This is to ensure smooth delivery of work products and reduce the need for excessive iterative cycles during the implementation phase to solve problems. The risks identified can be found in the Risk Assessment Plan.

Quantitative measurement of resource consumption: Estimates of the “Go Where GaiGai” project 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.

Regular reviews of project progress: Throughout the duration of the project, Group ONE shall meet weekly to review the progress of all project tasks, including management, planning, analysis, development, and testing. This is also a measure to ensure the workload throughout the project is balanced within team members.

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.