Project Plan

GuessWho!

Face Recognition Game to aid Dementia Patients

Delivered by:
Bookies
Joshi Chaitanya Krishna(U1522971F)
Genevieve Lam Wen Qi (U1521863H)
Sharma Vidur (U1522940D)
Yong Chen Feng (U1620913B)
Heng Zhi Guang (U1620660F)
Cheng Guanyu (U1621093D)

Submitted to –
Dr Althea Liang
Nanyang Technological University
School of Computer Science and Engineering

Table of Contents

1 Introduction

1.1 Overview

GuessWho! was designed as a web-based game to allow dementia patients to practice memory recall through face identification. The intended effect is for users to improve memory after engaging in the game.

1.2 Objectives and Scope

As a web-based application, GuessWho! seeks to be accessible to dementia patients without any specialized equipment, only requiring an internet connection and a smart device or computer.

In the game, users will be asked to match faces to the correct names. These faces will be faces of the friends or family of the user. Once the user has managed to successfully identify the faces, the caregiver can increase the difficulty by adding faces of famous people for face matching.

GuessWho! allows caretakers in charge of the patients to keep track of the patient's condition by keeping track of their progress. In the application, the user's scores will be recorded and the progression of the scores can be monitored. From this data, the caretakers will be able to quantify a patient's progress and develop appropriate treatment methods based on what has been observed.

2 Project Organization

2.1 Team Structure

The following is the list of executive roles of the project team:

- **Project Manager:** Chaitanya Joshi
- **Quality Assurance Manager:** Chen Guanyu
- **Quality Assurance Engineer**: Chen Guanyu
- **Lead Developer:** Heng Zhi Guang
- **Front-end Developer:** Genevieve Lam
- **Back-end Developer:** Yong Chen Feng
- **Release Engineer**: Vidur Sharma

2.2 Roles and Responsibilities

2.2.1 Project Manager: Chaitanya Joshi

- Oversees project progress
- Approves and executes project plan
- Allocates tasks and reports status of project to team members
- Manages team members and in charge of overall morale
- Represents the team to the outside world

2.2.2 QA Manager and Engineer

Quality Assurance Manager: Chen Guanyu

- In charge of overall product and process quality
- Responsible for preparation of quality documentation and reports
- Ensures software is of acceptable quality

Quality Assurance Engineer: Chen Guanyu

- Comes up with testing strategies
- Verifies software requirements align with user requirements
- Carries out test procedures
- Develops and manages test plans

2.2.3 Lead Developer: Heng Zhi Guang

- Overall technical lead
- In charge of the technical aspect of the product release
- Ensures system architecture is well designed

2.2.4 Front-End Developer: Genevieve Lam

- Ensures that the system design is technically feasible
- Ensures that user input is tested before being sent to the back-end
- Work together with back-end developer and lead developer

2.2.5 Back-End Developer: Yong Chen Feng

- Make use of detailed design document to implement working products
- Responsible for integration of user interface with server side logic
- Designs and implements data storage solutions
- Work together with front-end developer and lead developer

2.2.6 Release Engineer: Vidur Sharma

- Creates baseline and conducts release reviews
- Ensures application releases are delivered on time and within budget by measuring and monitoring progress
- Make use of Version Control Systems to coordinate release content and effort
- Builds and integrates changes for delivery

2.3 Team Communication

The proposed forms of communication within the team are as follows:

- Weekly meetings held on Tuesdays
- Google Drive will be used for file sharing between members
- Project updates and other meetings are announced in the WhatsApp group chat
- The team's Wiki page will be used for the uploading of finalized documents
- The team is divided into two subgroups; documentation and developers.

3 Process Definition

3.1 Lifecycle Model

The team intends to use the Agile model for software development throughout the project.

Because GuessWho! is designed to improve working memory of users, it is critical that GuessWho! is flexible and be able to change according to feedback from the patients and caregivers. This is especially the case as dementia patients have poor memory, which would mean that additional features may have to be added over time to combat this.

The Agile model would be adopted as it centred around the customer has a high level of risk management. If users feel that the application has become less satisfactory than previous versions, the application can then run on previous versions which meet the user's

requirements. In addition, every sprint produces a working application that the user can view, improving customer satisfaction.

The Bookies team intends to deliver the first iteration of functionality on the System Delivery date indicated in the Estimations section of this document. With further client interactions, newer functionalities may be added. Additional functionalities may be added if requested when meeting with the client.

3.2 Capability Maturity Model

The Capability Maturity Model consists of 5 levels: Initial (Level 1), Repeatable (Level 2), Defined (Level 3), Managed (Level 4) and Optimizing (Level 5). The target level for the software development process is Level 3 (Defined) as the team does not have the quantifiable data available to allow for objective analysis of the processes used.

Level 1 is the most basic stage of software development process where the processes conducted are ad hoc and chaotic. There are no process definitions. Although the product would be functional at the end, this usually comes at a high cost and the product may not be ready by originally planned date. In Level 2, the status of the project components and delivery of the services are made visible to the management at predetermined points. However, the team is of the opinion that Levels 1 and 2 are insufficient in ensuring a smooth implementation as both could result in the project exceeding its schedule.

Level 4 uses precise measurements to control the software development effort effectively. The team will need to be able to identify ways to identify and adjust the process to a particular standard without any measurable losses of quality or significant changes in the specifications. In Level 5, processes are constantly improved based on quantitative feedback from metrics measured in order to achieve established quantitative process improvement. As illustrated, Levels 4 and 5 are too complex and the team has neither the expertise nor the experience to implement such maturity levels.

Thus, Level 3 is the most suitable for this project given the team's current experience and expertise as it involves a well-defined and documented qualitative standard software process.

3.2.1 Key Process Areas

The Key Process Areas (KPA) for this Capability Maturity Model 3 include:

- Technical Solution
- Product Integration
- Verification
- Validation
- Decision Analysis and Resolution
- Integrated Project Management
- Organizational Process Focus
- Organizational Process Definition
- Risk Management

Due to the nature of GuessWho!, the following KPAs are not applicable:

- Inter-group coordination is not applicable since the implementation of this application involves only this team.
- Similarly, subcontract management process would not be required as the implementation of this application only involves this team.

3.2.2 Description of Key Process Areas

3.2.2.1 Technical Solution

Designing, developing and implementing solutions to improve the working memory of dementia patients who face difficulty in memory recollection make up the primary goal of the technical solution. Possible technical solutions include a bigger font size and larger buttons on the screen.

3.2.2.2 Product Integration

As multiple developers are working together in this project, the product integration must consider the programming style of each developer and ensure that no errors will be produced when the sub modules are integrated.

3.2.2.3 Verification

Users must be verified before they can begin using the application. This is to prevent any breaches in data security as well as the unauthorized access of confidential patient data.

3.2.2.4 Validation

The prototype demonstration has been included in the project plan. This is done to monitor the progress of the application and to determine if it is acceptable. In addition, this also seeks to make sure that the application is able to satisfy the requirements set out when placed in the actual environment to be used.

3.2.2.5 Decision Analysis and Resolution

In order to examine feasible decisions, a structured evaluation process will be used to compare alternatives with each objective in the project. The most appropriate solution will then be selected and implemented. In this process, a decision analysis and resolution map will be used for the analysis of alternatives.

3.2.2.6 Organizational Process Focus

Organizational structure process, process review and process improvement are three processes defined in this KPA. The purpose of this key process is to plan, implement and execute a process that facilitates improvement according to the skills and shortcomings of the team.

3.2.2.7 Integrated Project Management

The team has to involve relevant stakeholders in the project as external systems (such as a database) were included as part of the application. As such, it is necessary for the team to manage the project and to involve these stakeholders in a well described process that has been designed for the team.

3.2.2.8 Organizational Process Definition

A set of process assets has been created by the team. Process assets may include but are not limited to all recorded documents, templates, procedures of management planning and risk planning. The System Requirement Specifications (SRS), Change Management Plan, Configuration Management Plan and Risk Management Plan are what has been established. The team will work together according to the different deliverables.

3.2.2.9 Risk Management

Each time a decision is about to be made, risk management practices will carried out by the team. Each team member will attempt to identify the risks of carrying out a particular decision.

4 Project Management

4.1 Activity Dependencies and Schedule

ID	Task Name	Start	Finish	% Complete
1	**Project Planning**	**Mon 8/13/18**	**Thu 10/4/18**	**26%**
2	Preliminary Analysis	Mon 8/13/18	Fri 8/17/18	100%
3	System Analysis	Mon 8/20/18	Fri 8/24/18	100%
4	Conceptualize requirements	Mon 8/27/18	Fri 8/31/18	100%
5	Prepare project proposal	Mon 9/3/18	Fri 9/14/18	100%
6	Prepare project plan	Mon 9/17/18	Thu 10/4/18	0%
7	Prepare quality management plan	Mon 9/17/18	Thu 10/4/18	0%
8	Prepare risk management plan	Mon 9/17/18	Thu 10/4/18	0%
9	Prepare change management plan	Mon 9/17/18	Thu 10/4/18	0%
10	Prepare configuration management plan	Mon 9/17/18	Thu 10/4/18	0%
11	**Requirement Analysis**	**Fri 10/5/18**	**Fri 11/2/18**	**0%**
12	Initial use case diagram and description	Fri 10/5/18	Wed 10/17/18	0%
13	Prepare system requirement analysis	Thu 10/18/18	Fri 11/2/18	0%
14	**Design Phase**	**Mon 11/5/18**	**Fri 1/4/19**	**0%**
15	Database schema	Mon 11/5/18	Fri 11/30/18	0%
16	Prepare test plan	Mon 12/3/18	Fri 1/4/19	0%
17	Lo-fi user interface	Mon 11/5/18	Fri 11/30/18	0%
18	High-fi user interface	Mon 12/3/18	Fri 1/4/19	0%
19	**Implementation and Testing Phase**	**Mon 1/7/19**	**Tue 7/16/19**	**0%**
20	Application prototype	Mon 1/7/19	Fri 6/7/19	0%
21	Testing	Mon 6/10/19	Tue 6/11/19	0%
22	Demonstration on initial prototype	Wed 6/12/19	Thu 6/13/19	0%
23	Enhancement of prototype based on user feedback	Fri 6/14/19	Tue 7/16/19	0%
24	**Deployment and Maintenance**	**Wed 7/17/19**	**Fri 9/13/19**	**0%**
25	Prepare release plan	Wed 7/17/19	Thu 8/15/19	0%
26	Official deployment of application to client	Fri 8/16/19	Fri 9/13/19	0%
27	Maintenance	Wed 7/17/19	Fri 9/13/19	0%

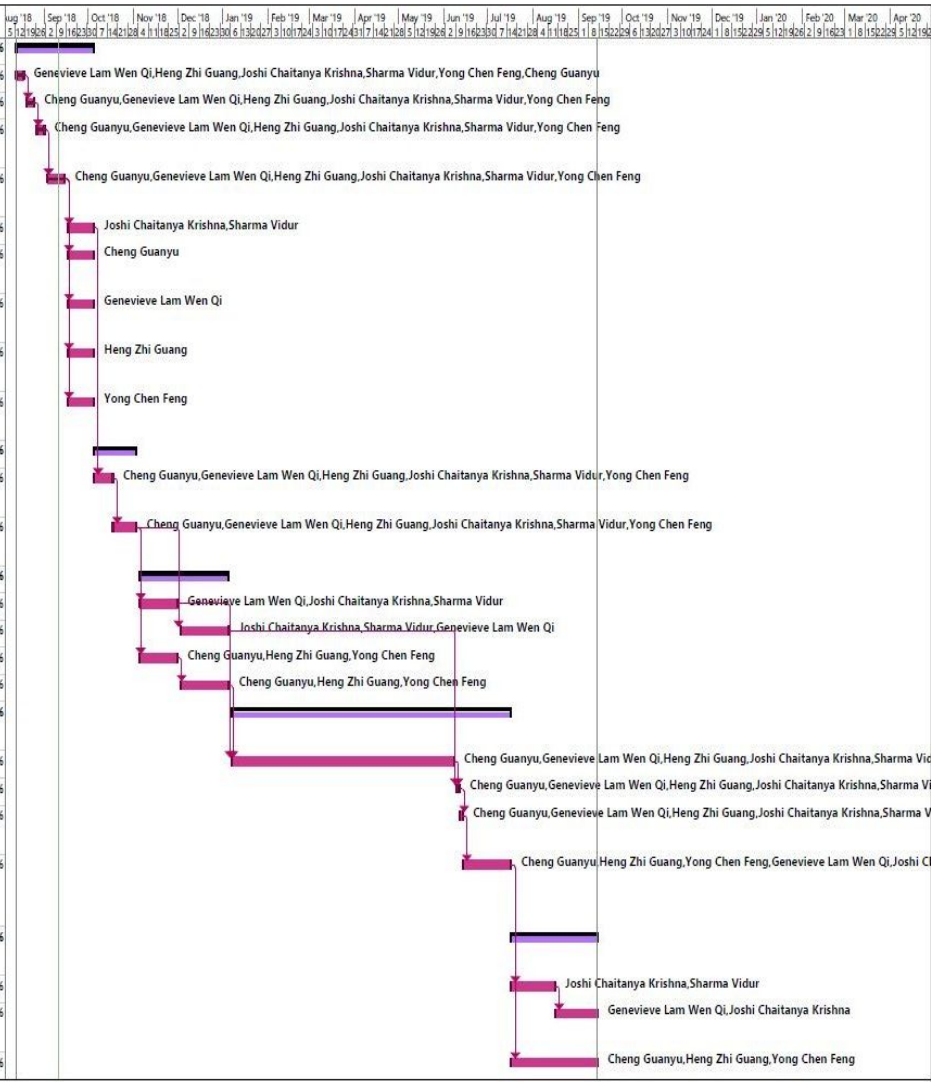

Figure 1: Project Schedule

4.2 Overview of Project Schedule

The entire project is split into 5 different phases: Planning, Requirement Gathering, Requirement Analysis, Design, Implementation and Deployment and Maintenance.

Phases	Duration
Planning	**5 Weeks** ● Information Gathering ● Discuss about framework ● Project Proposal
Requirement Gathering	**4 Weeks** ● Functional and Non-Functional Requirements ● Initial Use Case Diagram ● Define Data Dictionary ● Design User Interface
Requirement Analysis	**4 Weeks** ● System Requirement Specifications ● Quality Management Plan ● Finalize Use Case Diagram ● Use Case Description
Design	**7 Weeks** ● Application Skeleton **Planning – 5 Weeks** ● Project Plan ● Risk Management Plan ● Test Plan ● Configuration Management Plan ● Change Management Plan ● Release Plan
Implementation	**20 Weeks (Developing)** ● Prototype
Deployment and Maintenance	**5 Weeks** ● Deploy on client site ● Maintenance

4.3 Work Breakdown Structure

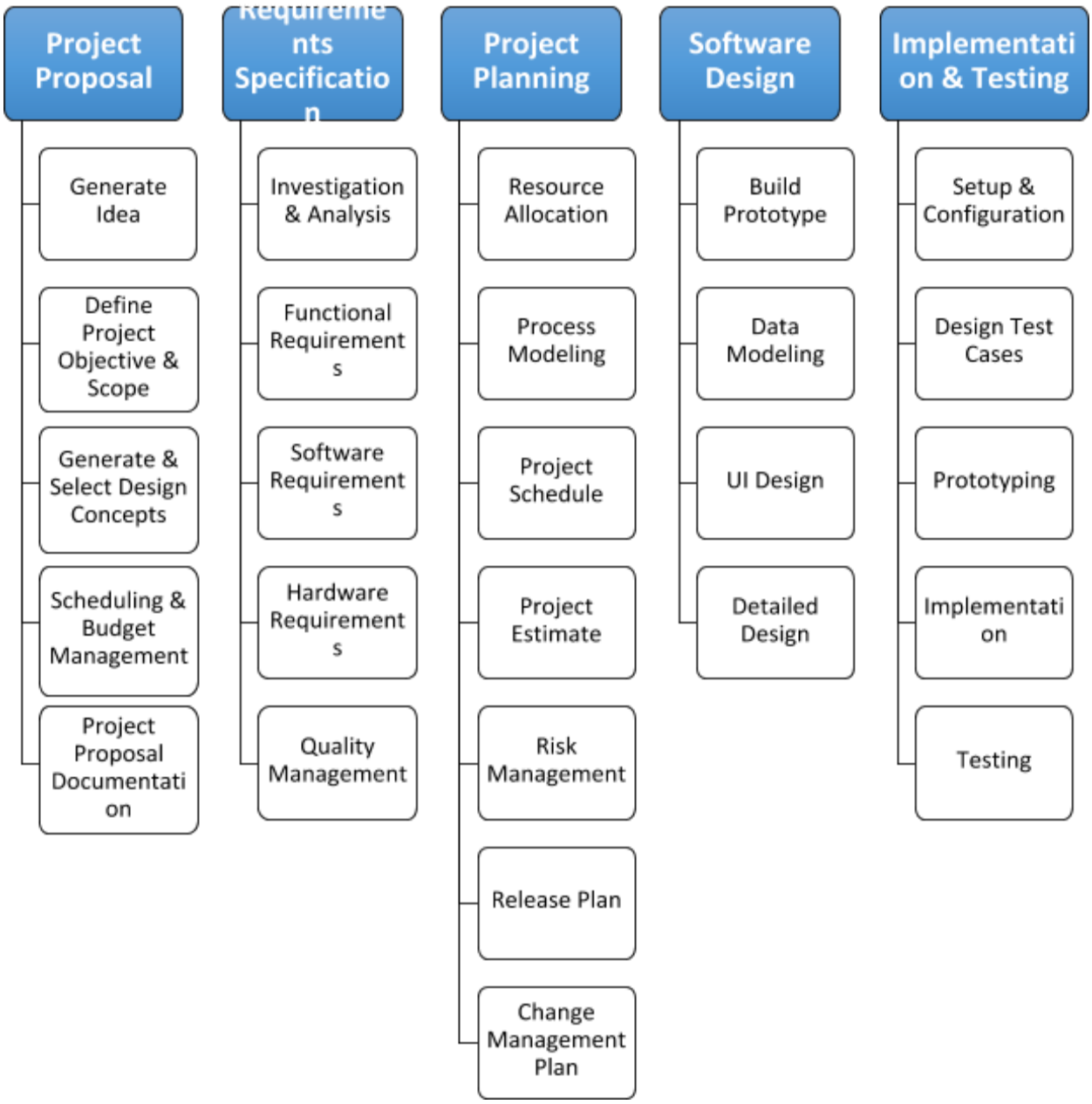

Figure 2: Work breakdown Structure

4.4 Work Packages

The project is divided into the important stages of the software development life cycle. They include the following:

- Project Proposal
- Requirement Specification
- Project Planning
- Software Design
- Implementation and Testing

4.5 Activity Dependencies

The following table presents the deliverable work packages and their dependencies:

Table 1: *Activity Dependencies of Work Packages*

Work Package #	Work Package Description	Duration	Dependencies
X01	Project Proposal	28 days	--
X02	Requirement Specification	28 days	X01
X03	Project Planning	21 days	X02
X04	Software Design	45 days	X03
X05	Implementation and Testing	210 days	X04

The following Activity Network Diagram presents a graphic visualization of the above in detail:

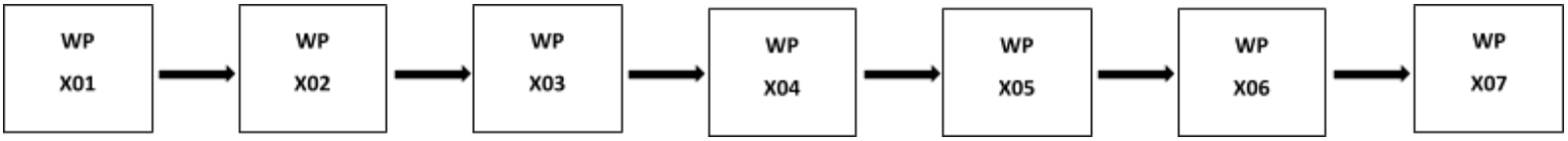

Figure SEQ Figure * ARABIC 3: *Activity Network Diagram of Dependencies*

4.6 Work Package Details

Details about the work packages are listed in the tables below. A team member indicated in bold is assigned as primary in-charge for each work package and will coordinate the completion of that package.

Table 2: WP X01 - Project Proposal

Project	GuessWho! Web Application
Work Package	X01 – Project Proposal (1 of 5)
Assigned To	Chaitanya, Genevieve, Chen Feng, Zhi Guang, Guanyu, Vidur
Effort	28 PD
Start Date	30/08/2018
Purpose	To determine concept and overview of the project which will be refined in the following work packages.
Inputs	None
Activities	This work package consists of creating a project concept and formulating an overview of the project, its objectives, and a set of proposed deliverables throughout the software development life cycle. The assignees are also responsible for drafting the discussions from group meetings into a formal report.
Outputs	A written document of the Project Proposal and Use Case.

Table 3: *WP X02 - Requirement Specification*

Project	GuessWho! Web Application
Work Package	X02 – Requirement Specification (2 of 5)
Assigned To	Chaitanya, Genevieve, Chen Feng, Zhi Guang, Guanyu, Vidur
Effort	28 PD
Start Date	13/09/2018
Purpose	To establish a common understanding between the patients' needs and the software project team of the customers' requirements addressed in the project
Inputs	Dementia Patients' Requirements
Activities	Identify the target audience, interview customers, write and build customer requirements.
Outputs	A written document of the System Requirement Specification (SRS) and Quality Plan

Table 4: WP X03 - Project Planning

Project	GuessWho! Web Application
Work Package	X03 – Project Planning (3 of 5)
Assigned To	Chaitanya, Genevieve, Chen Feng, Zhi Guang, Guanyu, Vidur
Effort	21 PD
Start Date	27/09/2018
Purpose	To estimate project schedule, allocate project resources and conduct risk management
Inputs	None

Activities	This work package involves allocating project resources by work breakdown, estimating time to completion of various activities and prototyping the application through UI diagrams.
Outputs	A written document of the Project Plan, Risk Management Plan and prototype demonstration

Table 5: WP X04 - Software Design

Project	GuessWho! Web Application
Work Package	X04 – Software Design (4 of 5)
Assigned To	Chaitanya, Genevieve, Chen Feng, Zhi Guang, Guanyu, Vidur
Effort	45 PD
Start Date	10/10/2018
Purpose	To determine high level architecture of application
Inputs	WP X01-X03
Activities	During this process, the team will decide on the layout of the game interface, and design the flow of application. The Lo-Fi prototype will be produced, along with other modelling diagrams such as architecture diagrams and class diagrams.
Outputs	High Level Design and Architectural Specification.

Table 6: *WP X05 - Implementation and Testing*

Project	GuessWho! Web Application
Work Package	X04 – Implementation and Testing (5 of 5)
Assigned To	Chaitanya, Genevieve, Chen Feng, Zhi Guang, Guanyu, Vidur
Effort	210 PD
Start Date	24/10/2018

Purpose	To develop and implement the application and create test cases for various components
Inputs	WP X01-X04
Activities	This work package includes implementing the prototype and creating unit and integrated tests for the application. The configuration management and test plans are also created in this work package.
Outputs	A written document of the Configuration and Change Management Plan, Test Plan and SVN repository of entire code of application.

5 Project Estimates

5.1 Code Size Estimation using Function Points

Calculations on unadjusted function points were based on the complexity of the functions found within the system. Code size is then subsequently estimated by adjusted function points.

5.1.1 Unadjusted Function Points

The system supports the following proposed Functional Requirements:

1. Patient:
    - The System shall allow Patient to play the game at any difficulty selected.
    - The System shall produce history of scores of Patient's account upon query.
    - The game shall reflect the scores of the Patient for that session after every round.
2. Caretaker:
    - The Caretaker shall view/modify all Patient accounts under its own account.
    - The System shall produce history of scores of any Patient account registered under Caretaker's account upon query.
3. Administrator:
    - The Administrator shall view/modify all accounts within System.
    - The Administrator shall upload photographs to update the game's database.
    - The System shall produce history of scores of any account registered within System upon query.

The measure of unadjusted function points is based off the five primary component elements: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each of these element falls in a range of Low, Medium or High Complexity.

The detailed evaluation of the complexity is as follows:

5.1.1.1 Rating Inputs

- Account Information (Username, Password, Contact Information)
- Game Difficulty Selection (Easy, Medium, Hard)
- Uploading photos to game's database

5.1.1.2 Rating Outputs

- Correct solution for each wrongly answered question
- Score of Patient after each game

5.1.1.3 Rating Inquiries

- Selecting Patient's account information from Caretaker's account
- Selecting answers from the options provided during the game

5.1.1.4 Rating Logical Files

- Accounts Database
- Photos Database

5.1.1.5 Rating Interface

- History of scores Interface
- Game Interface
- Login Interface
- Upload Photos Interface

The above details can be summarized as follows:

ELEMENT	DETAILS	COMPLEXITY
Inputs	Account Information	Medium
	Game Difficulty Selection	Medium
	Upload Photos to Game Database	Medium
Outputs	Correct Answer for Wrongly Answered Questions	Low
	Score of Patient after each Game	Low
Inquiries	Selecting Patient account information from Caretaker account	Low
	Selecting answers from options provided during Game	Low
Logical Files	Accounts Database	High
	Photos Database	High
Interfaces	History of scores Interface	Medium
	Game Interface	Medium
	Login Interface	Low
	Upload Photos Interface	High

Based on the table, the unadjusted function points is calculated as follows:

CHARACTERISTICS	LOW COMPLEXITY		MEDIUM COMPLEXITY		HIGH COMPLEXITY	
# INPUTS	**0**	0 X 3 = 0	**3**	3 X 4 = 12	**0**	0 X 6 = 0
# OUTPUTS	**2**	2 X 4 = 8	**0**	0 X 4 = 0	**0**	0 X 7 = 0
# INQUIRIES	**2**	2 X 3 = 6	**0**	0 X 5 = 0	**0**	0 X 6 = 0
# LOGICAL FILES	**0**	0 X 7 = 0	**0**	0 X 10 = 0	**2**	2 X 15 = 30
# INTERFACES	**1**	1 X 5 = 5	**2**	2 X 7 = 14	**1**	1 X 10 = 10
TOTAL UNADJUSTED FUNCTION POINTS	19		26		40	
	85					

5.1.2 Adjusted Function Points

The table below states the definition of each score that will be used for evaluation of adjusted function points:

SCORE	INFLUENCE LEVEL
0	No Influence
1	Insignificant Influence
2	Slight Influence
3	Average Influence
4	Significant Influence
5	Strong Influence

Based on Scoring Table, the adjusted function points are calculated as follows:

INFLUENCE FACTORS	DETAILS	SCORE
Data Communications	Application is more than a front-end, but supports only one type of TP communications	4
Distributed Functions	Data is prepared for transfer, then is transferred and processed on another component of the system, not for user processing.	2
Performance	Response time or throughput is critical during peak hours. No special design for CPU utilization was required. Processing deadline is for the next business cycle.	2
Heavily Used	Stated operational restrictions require special constraints on one piece of the application in the central processor or a dedicated processor.	2
Transaction Rate	High transaction rates affect the design, development, and/or installation phases.	3
On-line data Entry	More than 30% of transactions are interactive.	5
End-User Efficiency	Stated requirements for user efficiency are strong enough to require design tasks for human factors to be included.	4
On-line Update	On-line update of major internal logical files is included.	3
Complex Processing	The degree to which processing logic influenced the development of the application. No complex processing are present.	0
Reusability	Ten percent (10%) or more of the application code developed is intended for use in more than one application.	3
Installation Ease	No special considerations were stated by the user, and no special setup is required for installation.	0
Operational Ease	Start-up, back-up, and recovery processes were provided, but human intervention is required.	1
Multiple Sites	The needs of more than one installation site were considered in the design, and the application is designed to operate only under identical hardware and software environments.	1
Facilitate Change	Flexible query and report facility is provided that can handle simple requests. Business control data is kept in tables that are maintained by the user with on-line interactive processes, and the changes take effect immediately.	3
Total Score	**33**	
Influence Multiplier	**(Value Adjusent Factor)** Total Score x 0.01 + 0.65	**0.98**
Adjusted Function Points	Unadjusted FP x Influence Multiplier	**83.3**

5.1.3 Lines of Code

According to Capers Jones' firm, SPR, each Function Point requires 53 Source Lines of Code (SLOC) if the system is implemented in Javascript.

Therefore, **Total Lines of Code (LOC) = 83.3 FP x 53 LOC/FP = 4414.9 LOC**

5.2 Effort, Duration and Team Size Estimation

In order to estimate the effort and duration required for this project, both Top-Down estimation and Bottom-Up Estimation has been used. The combination of the two estimation methods can help the Bookies Team to deduce a reasonable and practical schedule for the project.

The estimates are also expanded to account for project management as well as the extra contingency time, which is allocated 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 estimations.

5.2.1 Top-Down Estimation

Top-Down Estimation uses Function Points as the basis to calculate Effort, Duration, Team Size, Compression Rate and Schedule.

Working Days	5 Days in a Week
Effort **= Size / Production Rate**	4414.9 LOC / (62 LOC/PD)* = 71.21 Person Day (PD) *LOC/PD statistics based on Industrial Benchmarks, 1997: Canada
Duration **= 3 x (Effort)$^{1/3}$**	3 x (71.21)$^{1/3}$ = 12.43 Person Day (PD)
Initial Schedule **= Effort / Working Days**	71.21 / 5 = 14.24 Weeks
Team Size **= Effort / Duration**	71.21 / 12.43 = 5.73 Person
Compression Rate **= Obtained Team Size / Actual Team Size**	6 / 7 = 0.86
Desired Schedule **= Initial Schedule x Compression Rate**	14.24 x 0.86 = 12.21 Weeks
Total Person Hours **= Effort x Daily Working Hours**	71.21 x 8 Hours = 569.68

5.2.2 Bottom-Up Estimation

In Bottom-Up Estimation, the estimation is based on the following factors:

- Average time taken to develop project is estimated at:
  - 15 Weeks Duration
  - 5 Working Days / Week
  - Total Days = 15 x 5 = 75 Days
- 7 Team members' working hours at 8 Hours / Week each
- Total Man-Hours = (Total Hours / Week) x Total Weeks = (7x8) x (75/7) = 600 Hours

5.2.3 Distribution of Effort

From the Top-Down and Bottom-Up estimation methods, the distribution of effort can be calculated as follows:

Based on the Top-Down Estimation as well as the Bottom-Up Estimation methods, the Distribution of Effort is calculated using the table below:

Components (1990s Industry Data)	Work Package	Distribution	Top-Down Estimation (569.68 Hours)	Bottom-Up Estimation (600 Hours)
Preliminary Design (18%)	Project Plan	9%	51.27	54
	Requirement Specification	9%	51.27	54
Detailed Design (25%)	User Interface	7%	39.88	42
	Technical Architecture	11%	62.66	66
	Data Modelling	7%	39.88	42
Code & Unit Testing (26%)	Code & Unit Testing	21%	119.63	126
	Online Documentation	5%	28.48	30
Integration & Testing (31%)	Integration & Quality Assurance	31%	176.6	186
Extrapolated Total Effort			569.68	600
2% For Project Management			11.39	18
3% For Contingency			17.09	12
Total Effort			598.16	630

Duration estimates within the table are based on the assumption that each and every Team member works an equal amount on any given work package during the full duration of the project.

5.3 Cost Estimates

The project's expenses can be broken down into the following table:

CATEGORY	EXPENDITURE NAME	RATE ($ PER UNIT / DAY)	QUANTITY (UNITS / NO. OF DAYS)	TOTAL
Selection Process	**Travel & Expenses**	$30	30 (Subjected to Gantt Chart)	$900
Implementation Process	**Private Git Repository**	$210 / Month	12	$2 520
	Servers	$350 / Month	12	$4 200
Labour Costs	**Project Manager**	$5 000 / Month	12	$60 000
	QA Manager	$4 500 / Month	12	$54 000
	Lead Developer	$4 000 / Month	12	$48 000
	Release Engineer	$3 500 / Month	12	$42 000
	Front-End Developer	$3 500 / Month	12	$42 000
	Back-End Developer	$3 500 / Month	12	$42 000
	QA Engineer	$3 500 / Month	12	$42 000
Contingency	**Contingency**	10% of Implementation Costs	-	$33 672
Maintenance	**Software**	$210 / Month	12	$2 520
	Hardware	$350 / Month	12	$4 200
Total Costs				$378 012

6 Product Checklist

Product	Estimated deadline	Place of delivery
Project Proposal	13/09/2018	Team Wiki
Use Case Model	13/09/2018	Team Wiki
System Requirement Specification (SRS)	27/09/2018	Team Wiki
Quality Plan	27/09/2018	Team Wiki
Project plan	18/10/2018	Team Wiki
Risk Management Plan	18/10/2018	Team Wiki
Prototype visualization	18/10/2018	Team Wiki
Code, video and documentation	01/11/2018	SVN
Design report on software maintainability	01/11/2018	Team Wiki
Configuration and change management plan	01/11/2018	Team Wiki
Release plan	01/11/2018	Team Wiki
Enhanced prototype	01/11/2018	SVN

Test plan	01/11/2018	Team Wiki
Test cases and Requirement Test Coverage Report	15/11/2018	Team Wiki
CMM level 2 definition	15/11/2018	Team Wiki
Documentation of Code	15/11/2018	SVN

7 Best Practice Checklist

Table 14: Best Practice Checklist

Practices	**Approved by/ Completed on**
All documentation must be in a standardized format, including ● Consistency in font and font color for header and body ● Line spacing/paragraphing style must be sensible ● Clear and easy-to-understand language must be used	GuessWho! team -
Requirement specification follows the following standard practices: ● Unambiguous language must be used ● Accurate, achievable and consistent requirements must be added ● Both functional and non-functional requirements must be added	GuessWho! team -

• Constraints must be stated and checked if requirements can be implemented within the known constraints. • Set of requirements that meets needs of stakeholders must be ensured • Complex functions must be avoided • Approval and acceptance of requirement by stakeholder must be ensured	
Modelling must follow the practices stated below: • Model elements must be properly labelled and include descriptions for each element • Descriptions must be clear and concise • Model must have high cohesion and loose coupling • No redundant definition of class attribute or methods must be ensured	GuessWho! team -
Design of the application must adhere to the practice stated below : • Design must take into consideration of future maintenance and extension • Design must be robust and able to handle errors • Design patterns must be considered • Design must be concise and simple • Adequate details must be provided for coding to begin • Design to be implemented must be direct and realistic	GuessWho! team -
While implementing the application, keep in mind the following practices:	GuessWho! team -

• Language and mechanism must be understood to implement the design • Test cases/unit tests must be documented for the code • Codes must have high cohesion and be loosely coupled • Code written must be easily understood by other developers	
When conducting testing, keep in mind the following practices. • Test cases must be designed to cover as many possible scenarios • There must be a test case to test each requirement • Test case must exist for boundary condition • Test case must exist to check for incorrect input • Stress and load test cases must be added	GuessWho! team -

8 Risk Management

Risk Management is covered in detail in a separated document called “Risk Assessment”. The document is available in the Team Wiki along with the Project Plan. It covers the plans and strategies to identify, control, and resolve risks that might occur during the process of development. It also covers the documentation and monitoring of such potential risks.

9 Quality Assurance

Quality Assurance is covered in detail in a separated document called “Quality Plan”. The document is available in the Team Wiki. Detailed guidelines and ideas are provided to ensure the process of quality assurance.

Many different points are discussed in the quality assurance, e.g. documentation, implementation, reviews, etc. Aspects such as standards and conventions are introduced and to

be maintained and fulfilled throughout the duration of the project. These aspects will guarantee good practice in specifications and implementation.

10 Monitoring & Control

Some of the measures used in the project to ensure the process of the product development include:

10.1 Quantitative measurement of resource consumption

The usage of resource is an important thing to consider. Improper resource consumption may lead to potential risks and problems. Such quantitative measurement is needed for the estimation of the resource required throughout the project.

10.2 Identification of major project risks

It is a good practice to identify early what major risks may arise throughout the course of the project. Initial identification helps the team to prevent potential risks and properly handle the problems. Some examples of major risks and preventive measures are discussed in the Risk Assessment.

10.3 Regular reviews of project progress

The GuessWho! team has SQ personnel to assess the project's risk management process and participate in weekly risk management meetings and report any software risks to the QAM and the project manager. The main communication mean is Whatsapp, used to discuss problems and changes. GitHub is used as the main repository to keep track of the changes and developments of the product, with version control. Google Drive is used as the cloud storage for relevant materials and documents. Lastly, the Team Wiki is used to store and arrange all documentations and report of the entire project. Final code developed shall be uploaded to the SVM repository in addition to GitHub.

10.4 Timeline Planning and task decomposition

The timeline planning is done according to the set deadlines of each deliverable and project milestone. Relevant tasks are broken down into sub-tasks to facilitate the timeline planning and workload distribution. The timeline and scheduled tasks are available in the Gantt Chart, where tasks and milestone can be set and the status of completion can be monitored. Even though the actual product development may not meet the ideal deadlines and milestones, the team will take the timeline planning seriously and try to minimize the differences. More details on the task decomposition are available in the Work Breakdown Structure (WBS).