

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

PreSchool GoWhere

Project Proposal for PreSchool GoWhere, an intelligent system that recommends preschools based on your child's needs.

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Andrew Teo

Alfred Chow

Ryan Tan

Team WannabeFAANG

Revision History

Version	Date	Author	Description	Approved By
1.0.0	02092022	Andrew Teo	Initial proposal development	Alfred Chow
1.0.1	03092022	Alfred Chow	Addition of contents required for Lab 1 submission	Andrew Teo
1.0.2	05092022	Ryan Tan	Addition of contents required for Lab 1 submission	Alfred Chow

Contents

Executive Summary	3
Statement of Problem	4
Objectives	5
Technical Approach	6
Customer Needs and Preferences	
Target Specifications	
Technology Consideration	
System Architecture/Platform	
Project Management	8
Team Structure	
Roles and Responsibilities of Team Members	
Deliverables	
Budget	
Communication and Coordination with Sponsor	
Team Qualifications	
Conclusion	13
Appendix A: Résumés of Team Members	14

Executive Summary

To all parents out there, are you facing difficulty in finding a suitable preschool for your children? Do you wish that there's a hassle-free solution that helps recommend preschools that cater to your child's needs? If so, our team hopes to address this burning issue with our web application, PreSchool GoWhere!

PreSchool GoWhere is a web application that aims to help parents find an ideal preschool for their child. We understand that every child has their own needs and preferences such as food intake and language used and parents themselves, might be busy and would prefer preschools that are nearer to them or provide transportation services. As such, PreSchool GoWhere aims to provide a platform where parents can quickly find a preschool that matches most of their needs and preferences. In addition, PreSchool GoWhere will be able to send a full detailed report on a particular preschool and parents can conveniently contact the preschool for any further enquiries.

Focusing on user-friendliness and having a natural user experience, PreSchool GoWhere will be designed as a convenient and efficient website for parents to perform customized searches for preschools in Singapore. PreSchool GoWhere is an all-day service where it provides support to parents 24/7.

Here at Team WannabeFAANG, we are a team of designers and developers that collaborate well together with a common objective and can develop a product with the highest quality within a specific timeframe and budget/cost.

Statement of Problem

It is common for parents in Singapore to face difficulties juggling between child care and their full time job, which can prove to be a stressful task. The difficulty of child care increases due to inflexible working hours. As such, parents will hope to find a quality preschool that not only ensures a safe environment for their child to learn and develop but also fits within their own preferences. However, finding a quality preschool within a short amount of time is easier said than done. This is highly attributed to a variety of reasons.

One of the main reasons is that every child has their own needs and preferences, making it difficult for parents to find a suitable preschool that caters to most of their child's needs. Take for instance, if their child requires the consumption of a certain type of food intake due to religion or is allergic to specific types of food, parents would first have to search up all preschools that offer this food type. Then, if the child also prefers to be in an environment that caters to his language type, parents would then again have to search for preschools that caters not only to his food, but also language type. As such, it can be very difficult for parents to continuously search for preschools that meet all their criterias. Hence, we aim to provide a hassle-free solution that takes in all the preferences of the child as inputs and finds the most suitable preschool that matches most of the child's preferences.

Another main reason as to why it might be difficult for parents to find a suitable preschool is that information of all preschools is not organized and might also be incomplete in various online sources. Hence, it might be tough for parents to collate all the information on various preschools before considering them individually. Moreover, some information might be lost in such a messy environment and parents might be distressed over spending large amounts of time researching. As such, PreSchool GoWhere aims to solve this issue and provide them with all the information that they need in an extremely short amount of time.

Objectives

The main scope of this project is to recommend suitable preschools to parents within a short amount of time and provide them with the relevant information of these preschools. The target audience for this system will be primarily parents who are finding a preschool for their children to attend. They will use a web browser to interact with our web application and obtain the list of recommended preschools and the relevant information of the preschools.

The design objectives are:

- (1) Recommends suitable preschools to parents based on their respective needs
- (2) Display filtered preschools on a visual map
- (3) Email a detailed report on the preschool to users upon request

The system will be able to filter among all available preschools based on a set of filters or preferences inputted by the user.

The System will be able to display preschools on a map using Google Maps Api to showcase the location of the preschool. Users will be able to click on any preschool on the map to view some information on the preschool

The system will be able to email a full detailed report on the preschool upon the user's request . The report consists of all the relevant information of the preschool such as location, language offered and contact details.

Technical Approach

The technical approach adopted to achieving the objectives mentioned in the proposal can be broken down into three different categories: Customer Needs, Target Specification and Technology. This will ensure that our application is developed to have a proper product market fit and accurately addresses the needs of our target audience. Furthermore, it provides the project team with a clear understanding of the best technologies to adopt to solve the problem.

Customer Needs

- (1) Parents need a website that will display information about all the available preschools in Singapore, such as address, opening days and hours
- (2) Parents need a website that will allow them to perform customized searches for preschools in Singapore.
- (3) Parents need a service which will allow them to save the relevant contact information about the targeted preschool.

Target Specifications

- (1) The website should allow parents to view the list of available preschools in Singapore.
- (2) Users should be able to easily create a custom search which will accurately return all available preschools that fit within their search criteria. The data retrieval duration should be fast, ideally 1-2 seconds wait time.
- (3) Upon finding a preschool that interests the user, they should have the option to opt to receive an email containing the relevant contact information about the targeted preschool.

Technology Consideration

Technology	Purpose	Remarks
ReactJS	Front End development	A declarative, efficient and flexible Javascript library that allows for building reusable UI components.
ExpressJS	Backend - Server Side development	A minimal and flexible NodeJS web application framework that robustly manages servers and routers.
NodeJS	Backend - Runtime Environment	An event driven Javascript runtime that supports server-side code in Javascript
mySQL/	Backend -	A relational database management system to store vast

PostgreSQL	Database	amounts of information.
Render, Microsoft Azure	Hosting and deployment	Cloud computing platform supporting a wide range of cloud services.
API Services	Data, Geolocation	API data fetching relevant information for project objectives.
Figma	UIUX design	Browser-based UIUX design application supporting features such as prototyping and code generation.

Project Management

In this project, Team WannabeFAANG will use the Agile methodology to implement the web application. As for group discussions, there will be group meetings held twice every week and the backlog will be updated using collaborative platforms such as Google Sheets or Trello. After each week, the team will use one of the group meetings to review the requirements and update the necessary changes if there is a need to do so. After defining the project scope, the product development process will be split into various phases such as planning, concept development, system-level design, detailed design, testing and refinement. The team then allocates the roles and responsibilities of the team members and team leader ensures that all team members work together and contribute to the project. This project will last for 3 months from September 2022 to November 2022.

Team Structure

- (1) Project Manager – PARASHAR KSHITIJ
- (2) Lead Developer – MALAVADE SANSKAR DEEPAK
- (3) Front-End Developer – ALFRED CHOW KENG YANG
- (4) Back-End Developer – DHANYAMRAJU HARSH RAO
- (5) QA Manager – ANDREW TEO SHAOMING
- (6) QA Engineer – TAN JIA HAO RYAN
- (7) Release Engineer – ALFRED CHOW KENG YANG

Roles and Responsibilities

- (1) Project Manager
 - Overall delivery of product
 - Ensures that deadlines are met
- (2) Lead Developer
 - Overall technical lead
 - Responsible for technical aspects of product release
- (3) Front-End Developer
 - Web application front-end programming
 - Participate in the entire SDLC
 - Generating work products including documentation, source code, unit and integration tests

(4) Back-End Developer

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

(5) QA Manager

- Ensures overall product and process quality
- Implementation of QA processes

(6) QA Engineer

- Devise test plans
- Perform testing on product

(7) Release Engineer

- Create baselines and build and integrate changes for delivery.
- Manage releases of prototypes.

Deliverables

Deliverables	Completion Date (Estimated)	Final Dateline
Use case model	7/09/22	8/09/22
Project Proposal	7/09/22	8/09/22
System Requirement Specification	20/09/22	22/09/22
Quality Plan	20/09/22	22/09/22
Project Plan	4/10/22	6/10/22
Risk Management Plan	4/10/22	6/10/22
Prototype Demo	4/10/22	6/10/22
Design Report on Software Maintainability	19/10/22	20/10/22
Configuration Management Plan	19/10/22	20/10/22
Change Management Plan	19/10/22	20/10/22
Release Plan	19/10/22	20/10/22
Presentation Slides	3/11/22	3/11/22
Test Plan	3/11/22	3/11/22
Test Cases and Requirements Test Coverage Report	3/11/22	3/11/22

Budget

In this section, we will be calculating all the necessary costs that this project will incur. The following table describes all the software tools that we need and their respective costs. The total cost amounts to **\$75,209**, as shown in the table below.

Item	Supplier	Quantity	Unit price	Cost/month	Total cost (3 months)
Database	Microsoft SQL server	1	\$209	-	\$209
Hosting Service	Render/ Microsoft Azure	1	-	-	-
Project Manager	-	1	\$5000	\$5000	\$15000
Team Member	-	5	\$4000	\$4000	\$60000
Total Cost					\$75209

Communication and Coordination with Sponsor

The primary means of communication and coordination with potential sponsors will be done via email, which not only is a formal means of communication, but also provides a formal record of all communicated information between the project group and relevant parties involved in the sponsorship.

Meetings will also be conducted with sponsors via teleconferencing platforms such as Zoom or Microsoft teams. These meetings will be conducted on a bimonthly basis, which is pivotal in creating a healthy communicative relationship between project members and sponsors. The project team will update the relevant sponsors of the project progress as well discuss any potential issues that have surfaced in the project that need additional support.

Team Qualifications

Ryan is an efficient and effective developer who can take an idea or practice and find new ways to improve upon it. Worked alongside Andrew and Alfred in creating a full-stack web application in another school module CZ2006.

Alfred is driven , diligent and dedicated to the completion of the assigned tasks.Worked alongside Andrew and Ryan in creating a full-stack web application in another school module.

Andrew is a solution-focused critical-thinker who defines objectives and sees them through to completion. Worked alongside Ryan and Alfred in developing a full-stack web application.

Sanskar is focused, determined and proactive while working on assigned tasks, as well as collaborative with other team members. He has worked on a full stack web application as a backend developer for the project 'MybikeSG' alongside Harsh and Kshitij.

Harsh is a determined and hardworking individual with a growth mindset. He worked on 'MyBikeSG' with Kshitij and Sanskar.

Kshitij is a dedicated and goal-oriented problem solver who believes in effective teamwork and resource management. He has prior experience in full-stack web development and cloud services. He has also previously worked with Harsh and Sanskar in CZ2006 - Software Engineering.

Conclusion

As finding a good preschool for their child is important for the child's growth, PreSchool GoWhere aims to be a valuable platform that would alleviate the troubles parents may face in finding a suitable preschool for their children.

Appendix A: Résumés of Team Members

The following pages present one-page resumes of the team members for this project.

Ryan Tan

Singapore | Email: 8j97w8ncp@mozmail.com |

Education

School of Computer Science and Engineering, NTU, Singapore

Aug 2020 – Present

Bachelor of Engineering (Computer Science), graduating in Aug 2024

Academic Projects

Foxtrot, CZ2006

- Housing web application for users to view past housing transactions, sellers to list flats and for users to contact sellers upon spotting a flat that interests them.
- Unique features of this web application include a simple facial recognition system that prevent users from setting a profile photo without a recognisable image. Users are also able to view nearby amenities based off a flat location.
- Tech stack: HTML5, CSS, Javascript, ReactJS, Express and Firebase.

Others

Written and Spoken Language: English and Mandarin

IT Skills: Word Document, Google Sheets, Google Docs.

Andrew Teo Shaoming
Email: andrewteo99shaoming@gmail.com

EDUCATION

Nanyang Technological University, Singapore Bachelor of Engineering (Computer Science)	Aug 2020 – May 2024 (Expected)
---	---------------------------------------

- Honours (Distinction)
- Relevant Modules: (1) Software Engineering (2) Data Structures and Algorithm (3) Probability and Statistics for Computing (4) Introduction to Computational Thinking & Programming

ACADEMIC PROJECTS

Nanyang Technological University, Singapore Multidisciplinary Design Project	Aug 2022 – October 2022(Expected)
---	--

- Developed a robotic system with diverse functionalities such as maze exploration and image recognition by utilizing the OpenCV library in Python for computer vision.
- Worked closely with the android development team to develop an android graphical user interface; Android user interface displays the robot's path and allows one to control the robot's movement.

LEADERSHIP & CO-CURRICULAR ACTIVITIES

Table Tennis Club in Tanjong Hall (NTU) Captain	Aug 2021 –Aug 2021
--	---------------------------

- Organized weekly table tennis sessions for residents of Tanjong Hall to train and improve their table tennis skills.
- Facilitated the Inter Hall Games competition for Table Tennis where student athletes would represent their respective halls to compete; Led a team size of 8 players to compete in various competitions across the year.

SKILLS & HOBBIES

Tennis, Table Tennis, Badminton, Billiard, Sketching, Designing

EDUCATION

School of Computer Science and Engineering, NTU, Singapore

Aug 2020 – Present

Bachelor of Engineering (Computer Science). graduating in Aug 2024

- Relevant Modules: (1) Software Engineering (2) Data Structures and Algorithm (3) Introduction to Computational Thinking and Programming

ACADEMIC EXPERIENCE

Multidisciplinary Design Project

- Developed a robotic system with various functionalities such as image recognition and maze exploration by utilizing OpenCV library in Python for computer vision
- Work on the RPI side of the system to enable communication between Android, Algorithm and Hardware by utilising multithreading in order to allow the system to function and achieve its functionalities

SKILLS

- Languages: Proficient in English and Mandarin
- Digital Skills: Python, MATLAB, C, C++, Java, Microsoft Office

HOBBIES & INTERESTS

- Reading, Sports (Frisbee)

DHANYAMRAJU Harsh Rao | Mobile No.: 9778 2392 | Email: HARSHRAO001@e.ntu.edu.sg

GitHub: HarshRaoD | LinkedIn: <https://www.linkedin.com/in/harsh-rao-dhanyamraju/>

EDUCATION

Nanyang Technological University, Singapore

Aug 2020 – Jun 2024 (Expected)

Bachelor of Engineering (Computer Science)

- Relevant Modules: (1) Data Structures & Algorithms, (2) Software Engineering, (3) Artificial Intelligence, (4) Intro to Data Science and Artificial Intelligence, (5) Intro to Databases

EXPERIENCE

Occutrack Medical (A Trendlines Medical Company)

May 2022 – Jul 2022

Machine Learning Intern

- Performed Time Series Analysis using ARIMA, SARIMA, GARCH, Autoformer, and ETSformer
- Explored Computer Vision Algorithms including CNNs, ViT, SWIN Transformers, STEGO
- Engineered applications with OpenCV, Mediapipe, Dlib, ORB, Yolo, and GANs
- Created Interactive dashboards with Plotly, Seaborn, Matplotlib, HTML, and CSS

ACADEMIC PROJECTS

Software Engineering - myBikeSG

Feb 2022 – Apr 2022

- Pioneered a web application to help cyclists find Bike Racks, Flat Routes, and Repair Shops close to current location or destination, deploying Agile Development
- Implemented with MongoDB, Express, React.js, and Node.js (MERN Stack) leveraging Google Maps API
- Hosted on Firebase (Google Cloud) and Heroku
- Collaborated using Git and GitHub

CO-CURRICULAR ACTIVITIES

IEEE NTU Student Branch

Sep 2021 – Jul 2022

Tech Officer

- Debugged iNTUition's virtual platform (South-East Asia's largest student-led hackathon)
- Conducted a workshop on Natural Language Processing (Techathelon Series)
- Coordinated a workshop on Graphs and Trees (Coding Nights)

SKILLS

Digital Skills: Python, Java, C, C++, SQL, Node.js, MongoDB, Data Science, React.js, Tableau, OpenCV

Languages: Proficient in English and Hindi, Conversant in Marathi

Soft Skills: Patient, Adaptable, Collaborative, Curious, Perseverant, and Hard working

Sanskar Deepak **Malavade** | Mobile No: +65 93723164 | Email: sanskard001@e.ntu.edu.sg
| GitHub: github.com/Sanskard001 | LinkedIn: linkedin.com/in/sdmalavade

EDUCATION

Nanyang Technological University, Singapore

Aug 2020 – May 2024

- **Bachelor of Engineering (Computer Science)**
- Course Modules: (1) CZ2006- Software Engineering (2) CZ2007- Introduction to Databases (3) BU5601 – Fundamentals of Management (4) HW0188 – Effective communication

SKILLS/INTERESTS

- **Cybersecurity** : Cyber threat landscape, security vulnerability and exploits understanding, types of malware attacks, system safeguards, network security devices, intrusion detection
- **Networking**: Network access, Internet protocol, network configuration, routing, transport, and application protocols.
- **Programming**: C++, Java, Python, JavaScript.
- **Soft Skills**: Patience, Collaboration , Verbal, and written communication
- **Digital**: Microsoft Word, Excel, PowerPoint

ACADEMIC PROJECTS

Nanyang Technological University

Object Oriented Design and Programming

Title: Restaurant Reservation and Point of Sale System

Oct 2021 – Dec 2021

- Developed a user-friendly restaurant point of sale system as a command line application in Java.
- Modelled a reservation system using UML class and sequence diagrams.

Software Engineering project

Title: MyBikeSG

Jan 2022- April 2022

- Developed backend to store bike rack information and cyclist accounts using MongoDB, Express, and Node.js .
- Designed an email-based One-time-Password system using Node.js for 2-factor authentication during signup.
- Deputy leader for a team of six members and ensured successful project completion within deadline.

HOBBIES & INTERESTS

Musical instruments (Tabla, Darbuka, Congo), Sports (Softball, Bowling, Swimming).

Kshitij PARASHAR | Mobile No.: +65 8925 1997 | Email: kp27doha@gmail.com
Github: [xitij27](https://github.com/xitij27) | LinkedIn: www.linkedin.com/in/xitijp

EDUCATION

Nanyang Technological University, Singapore Aug 2020 – May 2024
Bachelor of Engineering (Computer Science)

- Relevant courses: Software Engineering, Net Centric Computing, Introduction to Data Science and AI, OS.

DPS Modern Indian School, Qatar (CBSE Curriculum) Apr 2016 – Mar 2020

WORK EXPERIENCE

VFlowTech Pte. Ltd. May 2022 – Aug 2022

IoT Intern

- Reviewed MADS IoT platform and coordinated with developers to debug issues to improve user experience.
- Programmed firmware on Raspberry PI/Advantech using C++ with Modbus and HTTP/MQTT protocols.
- Evaluated and tested AWS, Google Cloud, Microsoft Azure and Digital Ocean services for IoT platform development.
- Spearheaded IoT Development team for creating a new platform using MERN stack.
- Programmed cloud servers using AWS EC2 and Digital Ocean Droplet.
- Engineered configuration of DNS and NGINX web servers along with TLS encryption.
- Studied IoT data to design predictive alerts for implementation of machine learning algorithms.

Mentogram

May 2021 – Aug 2021

Business Analyst Intern

- Examined coaching market to devise new strategy for growth and sales.
- Designed value proposition canvas and sales pitch by identifying pain points, goals and solutions to problems faced by coaching industry.

ACADEMIC PROJECTS

CZ2006 - MyBikeSG, NTU Feb 2022 – Apr 2022

- Pioneered a web application to help cyclists find Bike Racks, Flat Routes, and Repair Shops close to current location or destination.
- Created using MongoDB, Express, React, and Node.js (MERN Stack)
- Hosted application on Firebase (mybikesg-86ea4.web.app) and Heroku.

CO-CURRICULAR ACTIVITIES

Mission X Programme, Youth Corps Singapore May 2022

- Participated in workshops on planning impactful community service projects and effective volunteer management.

Computer Science and Engineering Club, Nanyang Technological University Oct 2020 – May 2022

Member of Academics Sub-Committee

- Moderated the '[Techterview 101](#)' event for helping students crack technical interviews.
- Solved and corrected solutions to Previous Year Examination Papers.

CANdroid Residential Education Programme, Hall 3 Feb 2022 – Mar 2022

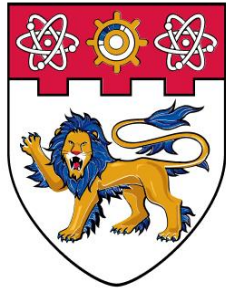
Student Assistant

- Conducted sessions and assisted participants in developing mobile applications using Android Studio.

SKILLS

Technical Skills: C/C++, Python, NodeJS, ReactJS, Java, SQL, Android Studio, HTML/CSS, Javascript, PowerBI, MongoDB, Microsoft Azure, AWS, Google Cloud, Digital Ocean, Fusion 360.

Languages: English (Proficient) and Hindi (Native).



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

PreSchool GoWhere

Use case model

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

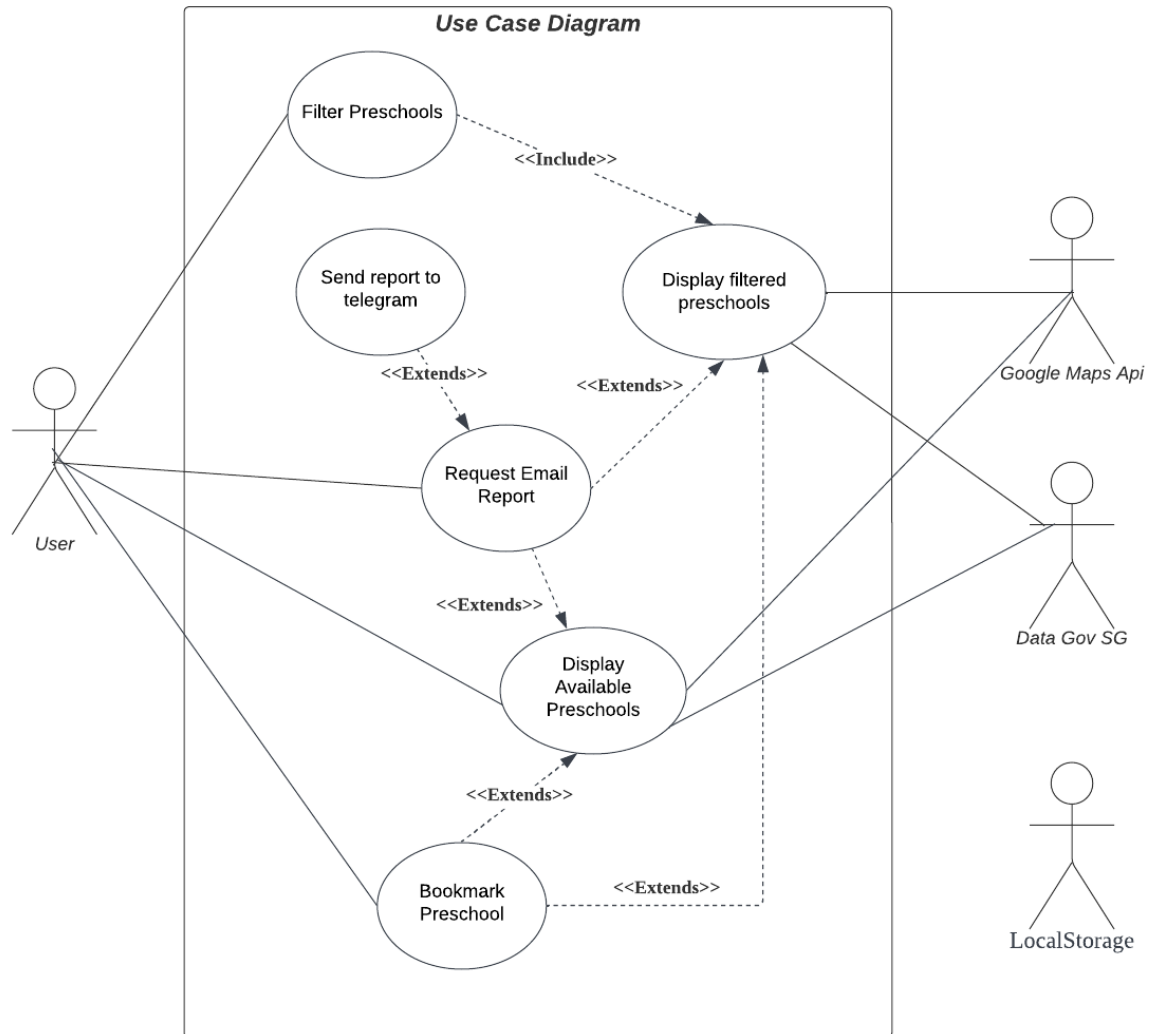
Alfred Chow

Andrew Teo

Ryan Tan

Team WannabeFAANG

Use Case Diagram



Use Case Descriptions

Use Case ID:	UC1		
Use Case Name:	Display all available preschools		
Created By:	Ryan Tan	Last Updated By:	Ryan Tan
Date Created:	04092022	Date Last Updated	04092022

Actor:	User
Description:	All available preschools in Singapore will be displayed on the user interface.
Preconditions:	1. Established connection to the internet via Mobile Data/ Wifi
Postconditions:	1. User is able to view information about each preschool displayed 2. Users are able to request for an email report containing the contact details of the preschool they are interested in.
Priority:	Medium
Frequency of Use:	High
Flow of events:	1. User opens website in browser 2. Website page launch, displaying all the available pre schools.
Alternative Flows:	-
Exception:	1. Sudden loss of internet connection
Includes:	-
Special Requirements:	-
Assumption:	-
Notes and Issues:	-

Use Case ID:	UC2		
Use Case Name:	Filter preschools		
Created By:	Ryan Tan	Last Updated By:	Ryan Tan
Date Created:	04092022	Date Last Updated	04092022

Actor:	User
Description:	Filters preschool search according to factors such as price, distance from home etc. The result is a displaying containing the preschools that fit into the search criteria.
Preconditions:	<ol style="list-style-type: none"> 1. Established connection to the internet via Mobile Data/ Wifi
Postconditions:	<ol style="list-style-type: none"> 1. Users are able to find preschools that fit into their search criteria. 2. User is able to view information of individual preschools that fit into their search criteria
Priority:	High
Frequency of Use:	High
Flow of events:	<ol style="list-style-type: none"> 1. User opens website in browser 2. User fills up the search bar with the criterias for their search 3. Website loads the search results, displaying only preschools that fit into the search criteria stated by the user.
Alternative Flows:	UC.2.AC.1: <ol style="list-style-type: none"> 1. User enters an irrelevant input that does not fit into the search criteria 2. Website will render an error notifying the user of the incorrect input.
Exception:	<ol style="list-style-type: none"> 1. Sudden loss of internet connection
Includes:	-
Special Requirements:	-
Assumption:	-

Notes and Issues:	-
-------------------	---

Use Case ID:	UC3		
Use Case Name:	Request Email Report		
Created By:	Andrew Teo	Last Updated By:	Andrew Teo
Date Created:	04092022	Date Last Updated	04092022

Actor:	User
Description:	Sends a report to the user's email, providing all the relevant information on the selected preschool.
Preconditions:	1. Established connection to internet via Mobile Data/ Wifi
Postconditions:	1. User is able to access his email and view the full report on all the relevant information of the preschool. 2. User can contact the preschool via email or phone for any further enquiries
Priority:	Medium
Frequency of Use:	High
Flow of events:	1. User opens the website in browser 2. User can choose to filter the preschools 3. User selects a particular preschool shown on the map. 4. User selects the option which sends a detailed report on the preschool to his email
Alternative Flows:	1. User's email is invalid.
Exception:	1. Sudden loss of internet connection
Includes:	-
Special Requirements:	-
Assumption:	-
Notes and Issues:	-

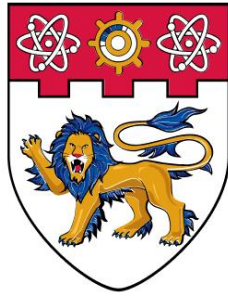
**** If we are implementing the bookmark functionality**

Use Case ID:	UC4		
Use Case Name:	Bookmark Preschool		
Created By:	Andrew Teo	Last Updated By:	Andrew Teo
Date Created:	04092022	Date Last Updated	04092022

Actor:	User
Description:	User is able to bookmark the preschool that he is interested in and revisit it anytime
Preconditions:	1. Established connection to the internet via Mobile Data/ Wifi
Postconditions:	1. User saves the interested preschool to his list of bookmarks 2. User is able to revisit all the preschools that was bookmarked previously
Priority:	Medium
Frequency of Use:	High
Flow of events:	1. User opens the website in browser 2. All available preschools are displayed to the user on a map. 3. User can choose to filter the preschools 4. User proceeds to bookmark a particular preschool. 5. The preschool is added to the list of saved preschools 6. User can revisit all the bookmarked preschools any time
Alternative Flows:	
Exception:	1. Sudden loss of internet connection
Includes:	-
Special Requirements:	-
Assumption:	-
Notes and Issues:	-

Use Case ID:	UC5		
Use Case Name:	Send Report to Telegram		
Created By:	Andrew Teo	Last Updated By:	Andrew Teo
Date Created:	04092022	Date Last Updated	04092022

Actor:	User
Description:	User will be able to request the report on preschool to be sent to his telegram account as a message
Preconditions:	1. Established connection to the internet via Mobile Data/ Wifi
Postconditions:	1. User is able to access his telegram and view the full message report on all the relevant information of the preschool. 2. User can contact the preschool via phone for any further enquiries
Priority:	Medium
Frequency of Use:	Medium
Flow of events:	1. User opens the website in browser 2. User can choose to filter the preschools 3. User selects a particular preschool shown on the map. 4. User selects the option which sends a detailed message report on the preschool to his telegram 5. User receives telegram message
Alternative Flows:	1. User's telegram account does not exist
Exception:	1. Sudden loss of internet connection
Includes:	-
Special Requirements:	-
Assumption:	-
Notes and Issues:	-



**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

PreSchool GoWhere

System Requirement Specifications (SRS)

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Andrew Teo

Alfred Chow

Ryan Tan

Team WannabeFAANG

1. Table of Contents

2. Problem Statement	4
3. Overview	4
3.1 Background	4
3.2 Overall Description	4
4. Investigation & Analysis Methodology	5
4.1 System Investigation	5
4.2 Analysis Methodology	5
5. Constraints	9
5.1 Real time updates of pre school data	9
5.2 Project Schedule	10
5.3 Proprietary hardware and software	10
6. Operational Requirements	10
6.1 Help Desk Support	10
6.2 Application Services and Technical support	10
6.3 System hardware fail over and routine back up	10
6.4 Audit Trail	11
7.1 User Account	11
7.2 View Available Preschools	11
7.3 Filter Preschools	12
7.4 Email Report	12
7.5 Bookmark Preschool	12
8. Non Functional Requirements	13
8.1 Security	13
8.2 Reliability	13
8.3 Performance	13
9. Input Requirements	14
9.1 User Account	14
10. Process Requirements	14
10.1 Data Gov SG information transfer	14
10.2 Data integrity	14
10.3 Data validation	14
10.4 Performance	14
10.5 Data repository	14
11. Output Requirements	15
11.1 Email Report	15
11.2 Saved Searches	15

12. Hardware Requirements	15
12.1 Network	15
12.2 Client Computers	15
12.3 Client Mobile devices	15
12.4 Production support systems	15
13. Software Requirements	16
13.1 Client Operating Systems (PC / Desktop)	16
13.2 Client Mobile Operating System	16
13.3 Web Application Compatible Browsers	16
13.4 Network system	16
13.5 Licenses	16
14. Deployment Requirements	17

2. Problem Statement

There are a vast number of preschools available in Singapore but searching for a suitable preschool remains a manual method of searching each school online individually for details. Despite the large amount of available preschools, there has yet to be a consolidated system or website for parents to search with set parameters that they desire or need. Thus, the searching process becomes tedious and long.

3. Overview

3.1 Background

In Singapore, there are more than 1800 preschool options available. With such a wide array of preschools to choose from, searching up each preschool individually on their websites can be very difficult. There are also instances where the preschool data cannot be obtained online easily, which would make the already tedious task even more unbearing.

At preschool, children build a strong foundation and pick up important social, pre-academic, and general life skills that will give them a leg up in school and beyond. As such, finding an ideal preschool that suits the needs of the child cannot be overlooked.

Knowing that it is a crucial age for the child's development, the lack of a proper website for the parents to browse and obtain accurate data can be a hindrance and adds up the work that the parents have to do.

In order to alleviate the trouble that the parents have to go through to give their child the most comfortable experience, a website that provides the interface of the government preschool database should be in place to aid the parents in finding a suitable preschool.

3.2 Overall Description

PreSchool GoWhere is a system that aims to improve the experience of parents in finding a suitable preschool for their child. Its main purpose is to provide the parents with all the data

they require about the respective preschool at their fingertips. PreSchool GoWhere has a web platform that enables the parents to choose set parameters that they want and to view the preschools that fit the parameters. In addition, the users will be able to obtain the contact details of preschool after they have filtered the parameters in order for parents to obtain more specific information about the preschool. Lastly, the web platform will be able to send a more detailed report to users that required the information. All of this will make the finding of preschool a easier process to go through for parents.

4. Investigation & Analysis Methodology

4.1 System Investigation

The PreSchool GoWhere system processes the registered email and the password to link the user account to the settings that the user has previously set. The system will then record the registered email and password and link the user to his registered account. The account will then have the user stored values, such as the preschools that were already bookmarked by the user on previous logins. The interface of the system will also read the information stored in the database and email the user the detailed reports of the preschool when user send in a request to the system. The system will then forward the details to the registered email of the account that the user is in.

4.2 Analysis Methodology

4.2.1 Feasibility study and requirements elicitation

To ensure that the targeted users, in this case the parents, are satisfied by the system, a development and implementation team composed of people knowledgeable about the issues will be organised and regular meetings to address these issues will be held regularly.

To ensure that the system is feasible and can be developed within the current time frame, a series of interviews with the managers and the developers of the PreSchool GoWhere system. Interview and feedback from the personnel and staff working directly with the system is also crucial and required to define the current environment and future system requirements.

To determine which requirements are the most appropriate based upon the results of the

interview, A Feasibility and Risk Assessment study will be done on all the requirements.

4.2.2 System analysis and requirements specification

❖ 4.2.2.1 Perform an analysis of the problem using object-oriented techniques.

An external view of the enterprise model will be formed via Unified Modelling Language(UML). The System Requirement Specification documents will form part of the documentation for the project.

Some of the desired features of the system include:

- The ability to filter desired preschools based on specific parameters
- The ability to display the filtered preschools
- System to generate a detailed report when requested
- System to send report via telegram
- System to send report via email
- The ability to bookmark specific preschools

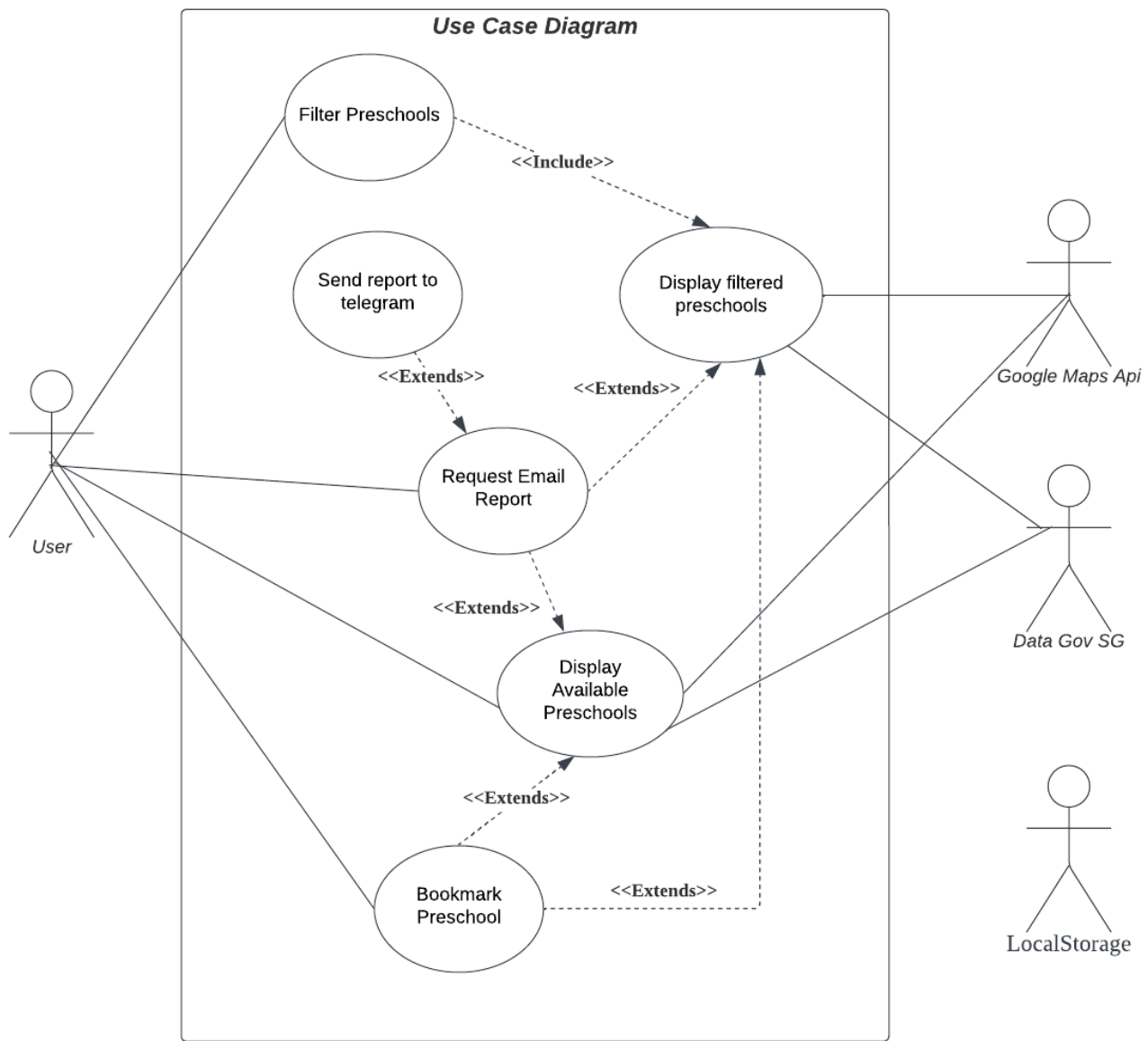
❖ 4.2.2.2 Scope and Limitations

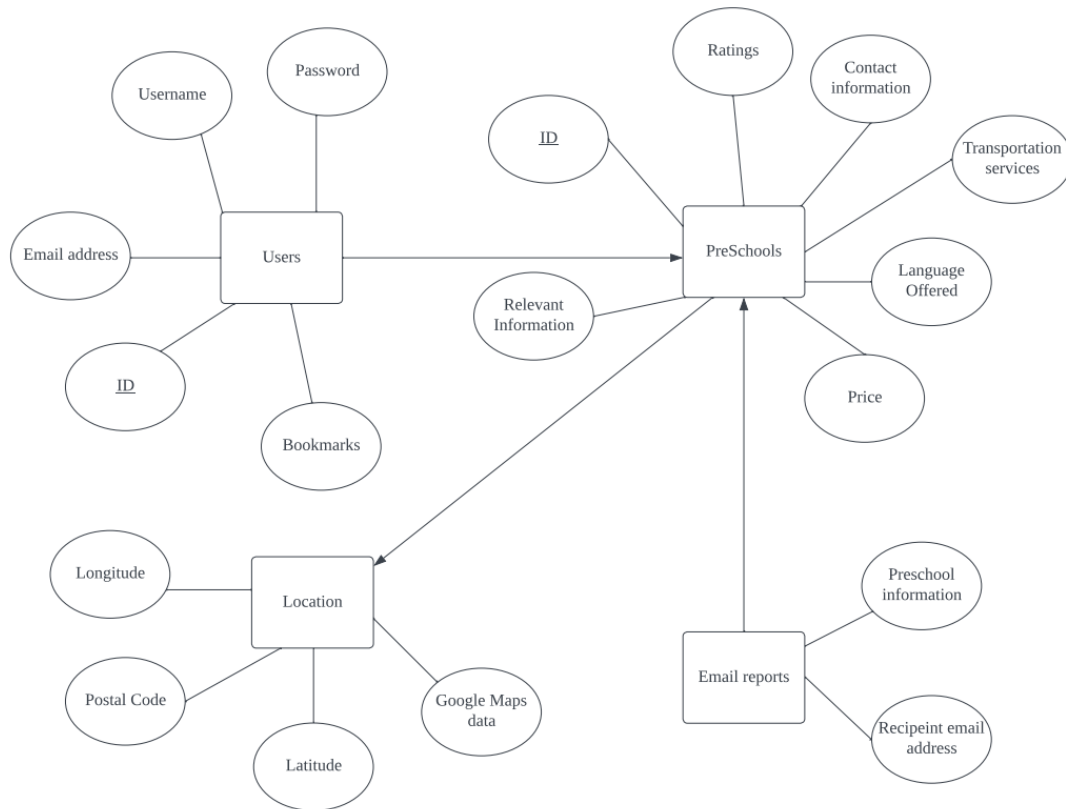
Analysis methodology will involve business analysis, requirement analysis, data analysis, process analysis, (web) and application architecture:

- Business analysis – State the business rules, business system interfaces, business function, business ownership, sponsorship and associated project budget requirement
- Requirement analysis – System I/O description, user requirement definition, functional and security requirement
- Data analysis – Involve data collection process, data validation, data storage, manipulation and retrieval
- Process analysis – Data/process flow analysis, process decomposition and system interfaces
- Application architecture – Analyse application information structure, usability, user interface design, interaction and application implementation.

❖ 4.2.3 Object-oriented design using UML

A detailed object-oriented design for the PreSchool GoWhere system will be developed. UML will be used again for the graphical representation and documentation of the design. The system will primarily concern itself with the searching process. At its core, a user will be able to choose his desired parameters of a preschool and the system will then retrieve information from the back-end system database that fits the chosen parameters. The system will then display the filtered preschools to the user. In addition, the system will also be able to forward the retrieved information of a particular preschool to the registered email of the user that requested the information. The use case will show how the user performs tasks on the website and the ER diagram will show how the data is mapped.





ER Diagram

4.2.4 Prototyping

The Object Oriented Rapid Prototyping (OORP) method will be used to implement a limited and functional prototype for the filtering and searching system. The prototype will be a working example of part of the system for demonstration and proof of concept purposes only. It will include web-based forms as an end-user interface with the data gov SG database. The prototype will be presented to the implementation team.

5. Constraints

5.1 Real time updates of pre school data

Newly updated data in the api providing data about pre-schools cannot be updated in real time as new data is fetched into the project's database via batch updates. The pre-school

data can be updated to its latest version on a scheduled basis.

5.2 Project Schedule

There is a three-month timeframe to implement a production system of an online data retrieval system from the project commencement in time for Fall 2022.

5.3 Proprietary hardware and software

Preschool Gowhere system is hosted on the internet via a cloud service provider, and therefore the system uptime relies heavily upon the operationality of the hardware and software of the cloud service provider.

6. Operational Requirements

6.1 Help Desk Support

For technical problems, such as system response latency, system downtime, user interface and account issues, system users have access to telephone assistance on a 24/7 basis.

6.2 Application Services and Technical support

Source code of application will be open-sourced on Github, this opens up the project to a vibrant ecosystem of open-source developers and white-hat hackers. An initiative like this will not only speed up the process of bug discoveries and fixes, but also help out in additional areas such as documentation of code for better understanding by other developers. Network Administrator and DBA support are also necessary to maintain a 24x7 system uptime and to efficiently administer any fixes and support in an event of system downtime.

6.3 System hardware fail over and routine back up

Computer operations center will handle system hardware tasks such as data tape back-up, hardware maintenance, fail over, scheduled system patches and maintenance.

6.4 Audit Trail

System audit trails are inherent part of all student registrations. Among others, all transaction records will capture what action was taken, when (time-stamp) the transaction occurred and who made the transaction.

7. Functional Requirements

The PreSchool GoWhere is a 24hr self-service web application that helps parents find a suitable preschool for their child based on their needs and preferences, hence providing a hassle-free solution for the. PreSchool GoWhere also has several other features such as allowing bookmarking of preschools and obtaining a full report on a particular preschool that the user is interested in.

7.1 User Account

1. The user must be able to create an account using their email
2. The user must be able to reset the password
3. The system must authenticate the user based on his email and password details

7.2 View Available Preschools

1. The user must be able to view all available preschools on a map
2. The preschools will be displayed on the map as pins with various colors, depending on their status such as bookmarked.
3. Upon clicking a particular pin on the map, there will be a small popup which displays the basic information of the preschool such as its name, location details and contact number.

7.3 Filter Preschools

1. The user must be able to input any filter conditions such as location, language offered, food type, operating hours, transportation services and ratings.
2. The system will be able to take in the user's filter conditions and filter out preschools that match the user's filters within 3 seconds.
3. Filtered preschools should be displayed on the map instead of all available preschools. Users can then click on any of the filtered preschools to view the basic information of the selected preschool.

7.4 Email Report

1. The user must be able to request for a full detailed email on a particular preschool that he is interested in
2. The system will be able to send a full detailed report to the user's email that was registered with PreSchool GoWhere. The detailed report, consisting of all relevant information on the preschool, will be sent to the user within 5 seconds.

7.5 Save Filters

1. The user will be able to save/bookmark a set of filter inputs in the filterbar.
2. The system will be able to save all of the user's filter options to the database.
3. The system must be able to load the filter options in the filterbar upon the user's request

8. Non Functional Requirements

8.1 Security

1. The system will be able to protect against unauthenticated and unauthorized users by requesting login information and user authentication.

8.2 Reliability

1. The system must be available and accessible 99.5 percent of the time.
2. The system must be able to handle input errors by the user. Errors such as invalid login information will be handled by the system and the system will prompt the user to re enter his details.
3. The system must be able to refresh in less than 3 seconds.
4. The system must be able to store data and backup the database daily.
5. The system will be able to prompt the user to check the internet connection if the wifi/4G from the user's device stops functioning halfway.

8.3 Performance

1. The system must be able to load within 2 seconds when the user enters the url link of the web application.
2. The system must be able to display any updated changes on the application if the preschool information or user related information changes.

9. Input Requirements

9.1 User Account

In order to use any of the features of our application, the user is required to sign up for an authenticated account using his email address, hence the unique identifier for each user account is the user's email address. This unique identifier is used to map all the user related information such as his bookmarks and the email address that the email report is being sent to.

10. Process Requirements

10.1 Data Gov SG information transfer

The system must be able to send and receive information to the Data Gov SG database system.

10.2 Data integrity

Information of the system must be the same on all interfaces

10.3 Data validation

Data error from the user's end and from the back-end database-processing end must be gracefully handled. There will be data validation and error-handling routines as part of the searching system.

10.4 Performance

The system must be able to handle a concurrent use on a 24/7 basis.

10.5 Data repository

The system will maintain the existing Data Gov SG database as the main repository of data.

11. Output Requirements

11.1 Email Report

Users must be able to receive a full detailed report sent to their email accounts by the system. The full detailed report on a particular preschool would consist of a summary of the preschool and all the relevant information with regards to the preschool

11.2 Saved Searches

Users must be able to bookmark any preschool that he is interested in and be able to revisit them anytime. Database will store a list of bookmarks made by the user while using our application and our application will display the list of bookmarked preschools to the user

12. Hardware Requirements

12.1 Network

University network infrastructure (wired and wireless)

12.2 Client Computers

Mac, Unix and Windows client computers

12.3 Client Mobile devices

Android and iOS client mobile devices

12.4 Production support systems

Web server computer(s) and related hardware support (back-up tapes, redundant drives, UPS, etc.)

13. Software Requirements

13.1 Client Operating Systems (PC / Desktop)

- Mac, Windows and Unix (any flavour)

13.2 Client Mobile Operating System

- iOS, Android operating system

13.3 Web Application Compatible Browsers

- React and Javascript compatible browser:
 - o Google Chrome
 - o Microsoft Edge
 - o Mozilla Firefox
 - o Safari
 - o Opera

13.4 Network system

Network software and protocols in order for systems to communicate:

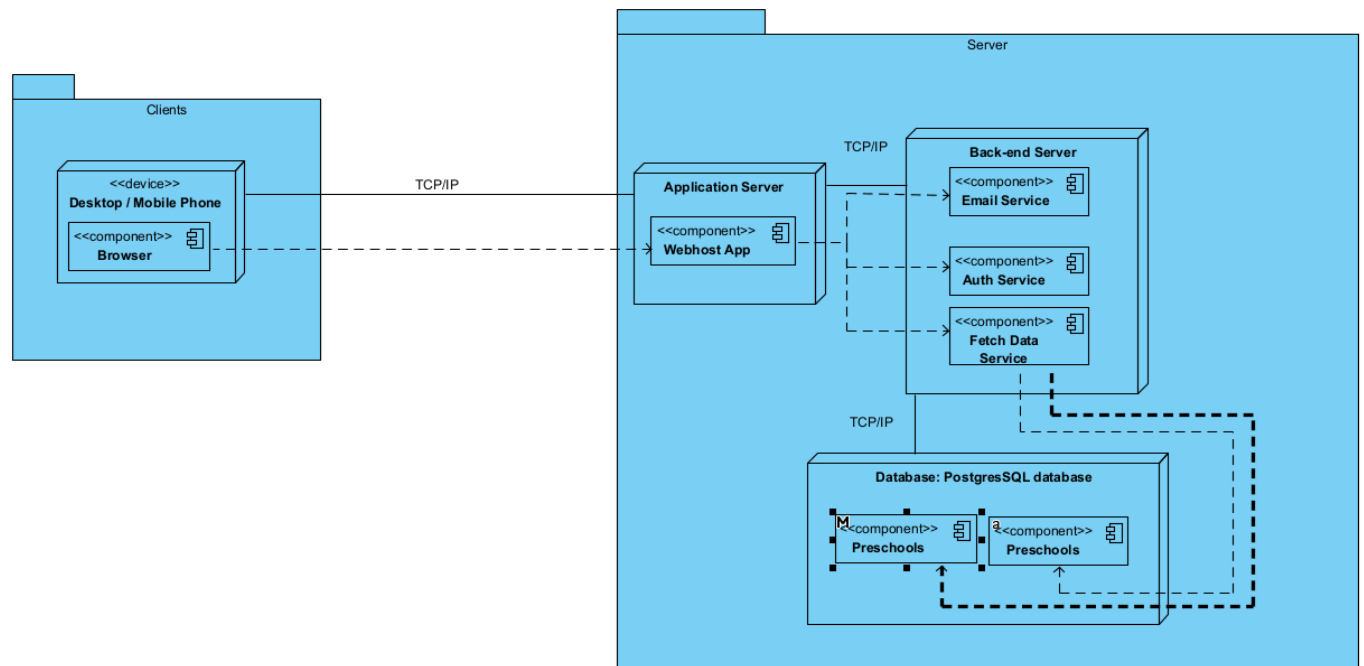
- TCP/IP
- HTTP
- HTTPS
- FTP

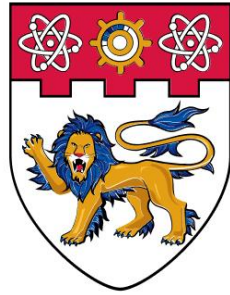
13.5 Licenses

Valid licenses are required to run software from third party vendors:

- To use application development tools (e.g. Integrated Development Environment)
- To use web server, application server and database software in development, test and production mode (e.g. NodeJS)

14. Deployment Requirements





**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

PreSchool GoWhere

Software Quality Assurance (SQA) Plan

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Andrew Teo

Alfred Chow

Ryan Tan

Team WannabeFAANG

Signature Page

	Name	Date
Prepared By:	Andrew Teo	10/09/22
Reviewed by 1:	Alfred Chow	21/09/22
Review by 2:	Ryan Tan	21/09/22
Approved by:	Alfred Chow	22/09/22

Document Change Record

Revision	Description of Change	Approved by	Date
0.10	Initial Template	Ryan	10/09/22
0.11	Update of content	Alfred	14/09/22
0.12	Update of content	Andrew	18/09/22
0.13	Update of content	Ryan	19/09/22
0.14	Format of document	Andrew	21/09/22

[illegible]

1. Purpose and Scope	7
1.1. Purpose	7
1.2. Scope	7
2. Reference Documents	8
3. Management	8
3.1. Management Organisation	8
3.1.1. Project Management	8
3.1.2. Assurance Management	9
3.2. Tasks	9
3.2.1. Product Assessments	9
3.2.2. Process Assessments	10
3.3. Roles and Responsibilities	10
4. Documents	11
4.1. Purpose	11
4.2. Minimum Document Requirements	11
5. Standards, Practices, Conventions and Metrics	11
5.1. Purpose	11
5.2. Software Quality Programme	12
5.2.1. Standard Metrics of ISO 9126	12
5.3. Practices	12
5.4. Conventions	12
6. Software Reviews	13
6.1. Purpose	13
6.2. Minimum Software Reviews	13
7. Test	14
8. Problem Reporting and Corrective Action	15
9. Tools, Techniques and Methodologies	17
9.1. Software Quality Tools	17
9.2. Software Testing techniques and methodologies	17
10. Media Control	18
11. Record Collection, Maintenance, and Retention	19
12. Training	20
13. Risk Management	20

1. Purpose and Scope

1.1. Purpose

The purpose of this Software Quality Assurance (SQA) Plan is to establish the goals, processes, and responsibilities required to implement effective quality assurance functions for the PreSchool GoWhere project.

The Software Quality Assurance Plan provides the framework necessary to ensure a consistent approach to software quality assurance throughout the project life cycle. It defines the approach that will be used by the QAM and Software Quality (SQ) personnel to monitor and assess software development processes and products to provide objective insight into the maturity and quality of the software. The systematic monitoring of products, processes, and services will be evaluated to ensure they meet requirements and comply with policies, standards, and procedures, as well as applicable Institute of Electrical and Electronic Engineers (IEEE) and ISO standards.

1.2. Scope

The purpose of SQA is to ensure that the software developed does not deviate from the original intended product. SQA is also concerned with identifying any errors, omissions, inconsistencies, and alternatives, enhancements or improvements that can be made at any stage of development.

PreSchool GoWhere is a full stack web application that allows users (parents) to find suitable preschools for their child, based on their child's needs and preferences. It aims to provide a hassle-free solution for parents to find an ideal preschool within a short period of time

This SQA plan covers the following primary features of the web application, namely:

1. **Login/logout** - This feature allows the user to log into the application to use its features. User identity is authenticated with the user records in our database.

2. **View Available Preschools** - This feature allows the user to view all the available preschools in Singapore on a map. Each preschool will be represented as a pin on the map and upon clicking any of the pins, it will display the preschool's basic information.
3. **Filter Preschools** - This feature allows users to input various filter conditions and then recommends a list of available preschools that matches the filters.
4. **Email Report** - This feature sends a detailed report to the user's email, which consists of all the relevant information on a particular preschool.
5. **Bookmark Preschool** - This feature allows the user to save any of the available preschools that he is interested in and revisit it anytime.

2. Reference Documents

- IEEE STD 730-2002, IEEE Standard for Software Quality Assurance Plans (http://standards.ieee.org/reading/ieee/std_public/description/se/730-2002_desc.html)
- ISO IEC 90003:2004 Software Standard (<http://praxiom.com/iso-90003.htm>)
- Project Plan
- System Requirement Specifications
- Project Proposal
- Use Case Model

3. Management

This section describes the management organisational structure, its roles and responsibilities, and the software quality tasks to be performed.

3.1. Management Organisation

The implementation of the quality assurance system is the responsibility of the Quality Assurance Manager (QAM).

3.1.1. Project Management

The Project Manager will be responsible for approving:

- The system requirement specification document
- The overall time scale for the project
- The choice of system development life cycle
- The choice of software development tools and techniques utilised
- The selection of project teams
- The training of project teams

3.1.2. Assurance Management

The QAM provides Project Management with visibility into the processes being used by the software development teams and the quality of the products being built. The QAM maintains a level of independence from the project and the software developers.

In support of software quality assurance activities, the QAM has assigned and secured Software Quality personnel from the pool of available SQ trainees to coordinate and conduct the SQ activities for the project and report back results and issues.

3.2. Tasks

This section summarises the tasks (product and process assessments) to be performed during the development of software. These tasks are selected based on the developer's Project Plan and planned deliverables, and identified reviews.

3.2.1. Product Assessments

The following product assessments will be conducted by SQ personnel:

The above listed processes will be assessed by the SQ:

- Project Proposal
- System Requirements Specification (SRS)
- Project Plan
- Risk Management Plan
- Initial Prototype

- Test Plan
- Configuration Management Plan
- Change Management Plan
- Release plan
- Login/Logout system
- Filter system
- Email Report system

3.2.2. Process Assessments

The following process assessments will be conducted by SQ personnel:

- Requirement Management
- Project Management
- Project Monitoring and Control
- Risk Management
- Test Management
- Software Configuration Management
- Release Management
- Software Development

3.3. Roles and Responsibilities

This section describes the roles and responsibilities for each assurance person assigned to the Project.

3.3.1. QAM

Responsibilities include, but are not limited to:

- Secure and manage SQ personnel resource levels
- Ensure that SQ personnel have office space and the appropriate tools to conduct SQ activities
- Provide general guidance and direction to the SQ personnel responsible for conducting software quality activities and assessments
- Assist SQ personnel in the resolution of any issues/concerns and/or risks identified as a result of software quality activities
- Escalate any issues/concerns/risks to project management

3.3.2. Software Quality Personnel

Responsibilities include, but are not limited to:

- Develop and maintain the project software quality assurance plan
- Generate and maintain a schedule of software quality assurance activities
- Conduct process and product assessments, as described within this plan
- Identify/report findings, observations, and risks from all software assurance related activities to the QAM

4. Documents

4.1. Purpose

This section identifies the minimum documentation governing the requirements, development, verification, validation, and maintenance of software that falls within the scope of this software quality plan. Each document below shall be assessed (reviewed) by SQ personnel.

4.2. Minimum Document Requirements

- Software Requirement Specification
- Quality Plan
- Project Plan
- Risk Management Plan
- Design report on software maintainability
- Configuration Management Plan
- Change Management Plan
- Release Plan
- Test Plan
- Test Cases and Requirements Test Coverage Report
- User Documentation
- Source Code Documentation

5. Standards, Practices, Conventions and Metrics

5.1. Purpose

This section highlights the standards, practices, quality requirements, and metrics to be applied to ensure a successful software quality program.

5.2. Software Quality Programme

In order to deliver a software that is of high quality, the project team has decided to adopt the ISO 9126 quality model, which is an international standard in evaluating the quality of the software product to be delivered.

5.2.1. Standard Metrics of ISO 9126

The ISO 9126 model comprises six metrics integral for the evaluation of the standard of the software. These metrics will be actively upheld, where information regarding each metrics will be collected, reported and maintained to create software quality assurance. SQ personnel will be involved in maintaining and reporting all relevant information about the evaluation to the Quality Assurance Manager.

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

5.3. Practices

The team will adopt the Agile Methodology model for the software development lifecycle. The use of the agile methodology during the project development phase allows for constant collaboration, iterated development as well as consistent testing of the software throughout the development phase. This will ensure that software quality will be consistently maintained at each progress of the software development, and improvements and fixes can be made in a quick and responsive manner.

5.4. Conventions

The project team understands that coding and development conventions are essential in not only reducing the cost of software maintenance, but an integral aspect of communication and understanding between developers of the project. As such, the team will be adopting the following coding and development conventions in the project:

- “PascalCasing” for classes and structures.
- Full caps for constants.
- “camelCase” for variables.
- Standardisation of indentation format and line length. This ensures that reviewers are able to easily narrow down on changes made in code during any form of code review (e.g. Pull Requests)
- Developers would need to create a production build before creating any pull requests for merge into the main development branch. A production build will help to ensure lower probability of the production site running into issues (e.g. crashes)
- Pre-commit hooks to alert on code quality issues (e.g. unused variables, unincluded dependencies) as well as any form of warnings that may stem from changes to code.

6. Software Reviews

6.1. Purpose

This section identifies the number and type of system/subsystem reviews and engineering peer reviews that will be supported by the SQ Personnel. The project milestone chart, and the SQ Personnel resource levels determine the reviews that are supported.

6.2. Minimum Software Reviews

The SQ team will ensure that development of software packages are inline to the specified criteria in every review iteration. The review content will be completely and accurately documented with sufficient details, capturing all Requests for Action, which will be reviewed and tracked to closure. In addition, SQ will assess the processes used to conduct the reviews to determine if appropriate personnel are in attendance, correct information is presented, entry and exit criteria are met, and appropriate documents are identified for update.

The following software reviews will be assessed by SQ:

- Project Plan Review
- Requirements Analysis Review
- Software Design Review
- Test Plan Review
- Acceptance Review
- Peer Review
- Code Review

7. Test

SQ personnel will assure that the test management processes and products are being implemented per Test Plan. This includes all types of testing of software system components as described in the test plan, specifically during integration testing (verification) and acceptance testing (validation).

The following types of test will be conducted for the software developed:

1. Unit testing

SQ personnel will be involved in testing individual methods, classes, components or modules of the software developed. The purpose of conducting unit test is to ensure that individual components and functionality of the software developed is functioning well and performing within the acceptable range of standards

2. Integration testing

SQ personnels will be involved in testing against a group of modules or components in the software developed. The purpose of conducting integration tests is to verify and ensure that different modules are able to work together with one another when put together. Conducting integration tests would also help in discovering possible bugs that may possibly exist when different modules work together so that the developer team will be able to address them.

3. Functional tests

SQ personnels will be conducting tests to verify if the software developed is inline with the business requirements of the application. Project requirements will be used as a standard metrics to evaluate whether the software performs as expected.

4. End-to-end tests

SQ personnels will be conducting test on the web application by replicating a user's behaviour in a complete application environment. End-to-end would involve testing whether the user flow when using the web application works as intended. It will also help to discover any issues pertaining loading of web pages when hosted on the internet as well as any other complex scenarios in using the web application.

SQ personnel will monitor testing efforts to assure that test schedules are adhered to and maintained to reflect an accurate progression of the testing activities. SQ will assure that tests are conducted using approved test procedures and appropriate test tools, and that test anomalies are identified, documented, addressed, and tracked to closure. In addition, SQ will assure that assumptions, constraints, and test results are accurately recorded to substantiate the requirements verification/validation status. SQ personnel will review post-test execution related artefacts including test reports, test results, problem reports, updated requirements verification matrices, etc.

8. Problem Reporting and Corrective Action

The SQ team has discussed and listed out a set of foreseeable problems which may arise over the course of the software development.

The following problems are:

1. Development
 - a. Software functionalities developed are not in tandem with the defined business requirements.
 - b. Incorrect behaviour of application functionality
 - c. Deviation from coding and development conventions
2. Documentation
 - a. Incompleteness and lack of core content of documentation
 - b. Incorrect docs content
 - c. Content in documentation is outdated and not in line with new updates in software
 - d. Spelling errors in documentation
 - e. Documentation is inconsistent with application developed
 - f. Formatting issues

SQ personnel generate, track, and trend assessment findings and observations in a centralised Reporting and Corrective Action System. The SQ team will be using both Trello, Jira and Google Sheets for the Reporting and Corrective Action System.

The SQ team will be adopting the following Standard Operating Procedures in managing corrective actions:

1. SQ personnel are to conduct regular reviews and scheduled checks to discover and identify potential problems in software.
2. Upon discovery of a problem, SQ personnel will evaluate the severity of the issue and priority of fixture before notifying the QA Manager and Project Manager of the issue.

3. Project Manager and Lead developer will re-evaluate the severity of the issue and priority of fixture.
4. If necessary, other relevant parties will be notified (e.g. Front end developer, release engineer) and a meeting will be scheduled to discuss the issue.
5. Through the discussion, a consensus, solution and deadline for correction will be determined.
6. All relevant details pertaining to the issue will be logged on the google sheets for Reporting and Corrective Action system. Status of the issue as well as any further communication between relevant parties will be done via Jira and Trello.
7. After the correction has been implemented, SQ team will review the corrective actions and perform the necessary evaluation tests again.
8. If the issue has been addressed following the predefined corrective actions to be taken, it will be marked as resolved on all relevant tools used for corrective actions. Else, the team has to revert to step 3 again.

9. Tools, Techniques and Methodologies

SQ personnel will require access to the following:

9.1. Software Quality Tools

- Microsoft Office tools (i.e., Word, Excel, and PowerPoint)
- Trello, Jira (For tracking of status of software bugs as well as SQ workflow support)
- Selenium, Jest, Lighthouse for Google Chrome (automated web application testing)
- Modern Web Browsers (i.e., Google Chrome, Mozilla Firefox, Safari)
- Postman (api testing tool)

9.2. Software Testing techniques and methodologies

SQ personnels will be involved in both black and white box testing as a means to maintain software quality via the discovery of issues that need to be fixed. The issues discovered during the execution of these tests will be addressed according to the ranking system of **bug severity** and **priority**.

9.2.1. Black Box Testing

Software developed will be tested on an external level, where implementation and design of any forms of functionality is unknown to the SQ personnel. Details of externally visible software bugs will be taken down by the SQ personnel and relayed to the development team for fixes

9.2.2 White Box Testing

Software developed will be tested under the knowledge of the design and implementation of the code. SQ personnel will be involved in testing control flow, code usability and readability, input and expected output, as well as sieve out any potential security loopholes present.

9.2.3 Automated tools for software testing

SQ personnels will leverage on automated software testing tools during the process of testing to increase both the efficiency and effectiveness of software testing. As the size and complexity of the software grows throughout the software development cycle, automated software developments tools will be critical and heavily relied upon in supporting performance of regression testings, unit testings and end-to-end testings

9.2.4 Agile Methodology for software quality

SQ personnels will be involved in the software development lifecycle and will adhere closely to the Agile Methodology principle. SQ personnels will be required to actively communicate with the development team on any issues regarding the software quality, as well as to track the progress of the fixes in software quality issues.

10. Media Control

SQ deliverables will be documented in one of the following Microsoft software applications: Word, Excel, or PowerPoint. Deliverables will be in soft copy, with the exception of completed checklists from process and product assessments. See Section 12 for additional details on the collection and retention of key records. Software Quality personnel will request space on the project's secured server for SQ records. This server is password protected and backed up nightly.

The following services are used for this project for media control:

1. MediaWiki
2. Git and Github
3. Google Drive

MediaWiki is a popular free server-based software, which is powerful, scalable and feature-rich. It is beginner-friendly as it uses MediaWiki's wikitext format, thus users can

edit them easily without any knowledge of HTML or CSS. MediaWiki keeps track of every edit made and every edit change is reversible and can be returned to previous versions. It is also able to manage any images and multimedia files uploaded to the MediaWiki server,

Git is a free to use, open-source version control system that is installed on the local system and available for branching. GitHub is a cloud-based Git repository hosting services, which allows for sharing of codes, in turn allowing for collaboration between members. Revision and editing of code is also possible through GitHub.

Google Drive is used for collaboration for all documentation, as it has a built-in office suite (Google Docs, Google Sheets), which allows for real-time collaboration and version control. Google Drive is also for file storage and file sharing.

11. Record Collection, Maintenance, and Retention

SQ personnel will maintain records that document assessments performed on the project. Maintaining these records will provide objective evidence and traceability of assessments performed throughout the project's life cycle. There are two types of records that will be maintained: Hardcopy and Electronic. SQ personnel will maintain electronic or hard copies of all assessment reports and findings. SQ Project folders will contain hardcopies of the assessment work products such as completed checklists, supporting objective evidence, and notes.

The table below identifies the record types that will be collected, as well as the Record Custodian and Retention period

Record Title	Record Custodian	Record Retention
SQA Assessments	SQ Personnel	One Year

SQA Checklists	SQ Personnel	One Year
Deliverable Defects	SQ Personnel	One Year

12. Training

SQ personnel have fundamental knowledge in the following areas through prior experience, training, or certification in methodologies, processes, and standards:

- Audits and Reviews (Assessments)
- Risk Management
- Software Assurance
- Configuration Management
- Software Engineering
- ISO 9001, ISO 9000-3
- CMMI
- Verification and Validation

13. Risk Management

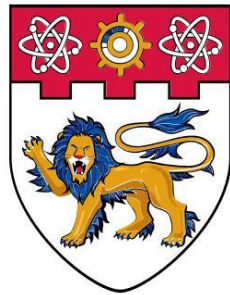
SQ personnel will assess the project's risk management process and participate in weekly risk management meetings and report any software risks to the QAM and the project manager.

With regards to the Software Quality Plan, the SQ team has identified an additional risk that has not been accounted for = Time Mismanagement. As such, a risk assessment has been done to mitigate it.

Risk	Risk Category	Risk Probability	Project impact	Trigger of Risk	Risk Strategy	Risk Response	Risk Zone	Risk Owner
Time Mismanagement	People	High	High	-Underestimating the complexity of the project -Overconfidence of capabilities	Mitigate	-Conduct regular meetings to ensure deadlines are met -Project team may need to work overtime to meet deadlines	High	Quality Manager

14. SQA Plan Change Procedure and History

SQ personnel are responsible for the maintenance of this plan. It is expected that this plan will be updated throughout the life cycle to reflect any changes in support levels and SQ activities. Proposed changes shall be submitted to the Quality Assurance Manager (QAM), along with supportive material justifying the proposed change.



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

PreSchool GoWhere

Software Quality Assurance (SQA) Plan

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Andrew Teo

Alfred Chow

Ryan Tan

Team WannabeFAANG

REVISION HISTORY

Version	Date	Author	Description of Changes	Approved By
1.0	1092022	Ryan Tan	Initial Project Plan structure	Andrew Teo
1.1	13092022	Andrew Teo	Added lifecycle mode	Ryan Tan
1.2	17092022	Alfred Chow	Added Risk management content	Ryan Tan
1.3	25092022	Ryan Tan	Added code estimate	Andrew Teo
1.4	26092022	Andrew Teo	Quality Assurance, Monitoring and Control	Ryan Tan
1.5	28092022	Alfred Chow	Activity Dependencies and Schedule	Ryan Tan

1 Introduction	2
1.1 Project Overview	2
1.2 Project Description and Scope	2
2 Project Organisation	3
2.1 Team Structure	3
2.2 Roles and Responsibilities	3
2.3 Team Communication	4
3 Process Definition	5
3.1 Lifecycle Model	5
4 Schedule	6
4.1 Activity Dependencies and Schedule	6
4.2 Work breakdown Structure	7
4.3 Work Packages	8
4.4 Activity Dependencies	8
5 Project Estimates	9
5.1 Code Size Estimation using Function Points	9
5.1.1 Unadjusted Function Points	9
5.1.2 Adjusted Function Points	10
5.1.3 Lines of code	12
5.2 Efforts, Duration and Team Size Estimation	13
5.3 Cost Estimates	14
6 Product Checklist	15
7 Best Practice Checklist	16
8 Risk Management	17
9 Quality Assurance	18
10 Monitoring and Control	19

1 Introduction

1.1 Project Overview

Preschool GoWhere is an informative website that aims to provide parents with an effective and convenient service for looking up preschools located in Singapore. PreSchool Gowhere enables users to perform customised searches for preschools in Singapore tailored according to their needs, such as location. This will help to ensure that users are able to search for preschools which fits into their needs.

1.2 Project Description and Scope

PreSchool GoWhere is a search tool for effectively looking up for preschools in Singapore. It is based on the preschool centers data released on Data.gov.sg. Data is stored on a cloud based PostgreSQL, and will return results according to the user's queries.

The target audience for this product will be mainly parents who are looking for a suitable pre school to enroll their children in. Parents will be able to access the website through our web interface using any device that contains a web browser. They will then be able to view the various pre schools in Singapore, perform customised searches and bookmark preschools .

The design objectives are listed as:

- (1) List all preschools in Singapore
- (2) Customised Search for preschools
- (3) Bookmarking of preschools
- (4) Email service for contact information of preschool

2 Project Organisation

2.1 Team Structure

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

- Project Manager: Parashar Kshitij
- Lead Developer: Malavade Sanskar Deepak
- Back End Developer: Dhanyamraju Harsh Rao
- Front End Developer: Alfred Chow
- Quality Assurance Manager: Andrew Teo
- Quality Assurance Engineer: Ryan Tan
- Release Engineer: Alfred Chow

2.2 Roles and Responsibilities

Project Manager: Kshitij

- Oversees the progress of the project, scheduling of software development timeline
- Approval and execution of project plan and additional features to be added
- Assignment of tasks and reports status of project to team members
- Manage and motivate team
- Represents the team in communication with relevant stakeholders

Lead Developer: Sanskar

- Overall Technical Lead
- Designing of technology stack to be used for project
- Review and merge pull request made by team members
- Reports to project manager development team progress
- Coordinates with developers to track progress of project, address any concerns that arise
- Ensure effective communication between team members
- Security of software product

Front End Developer: Alfred

- Implementation of visual elements of PreSchool Gowhere website
- Front End optimisations of website to ensure that it is both scalable and performant

- Front end user input validation (E.g. Login, Signup)
- Design and develop high quality UI/UX experience for users.

Back-End Developer: Harsh

- Handling of server side application logic
- Design and development of backend database
- Communication with FrontEnd developer to integrate work
- Handling of backend security

2.3 Team Communication

Team WannabeFAANG communication channel include the following:

- Weekly meetings held during weekends on Saturday or Sunday
- Telegram group for discussion of project, or to receive updates from project manager
- Collaborative Workspace using Trello, to easily communicate about specific issues with regards to development of software
- Project group is further splitted into a few subgroups, each focusing on their own assigned task.

3 Process Definition

3.1 Lifecycle Model

Team WannabeFAANG intends to use the Incremental Development Model throughout the development of PreSchool Gowhere. An Incremental Development Model provides more flexibility than the traditional Waterfall SDLC as teams will go through an iterative process of development, improvement, unit testing, integration and quality assurance. Though incremental development and improvement, the quality of the product will be improved alongside the development of new features of the project and bug fixes can be addressed as soon as possible. Coupled alongside iterative unit testing and quality assurance, bugs pertaining to new or existing features developed can be rapidly addressed. The team will also be able to ensure production readiness of the product due to the continuous integration process. This will reduce the risk of the team being unable to deliver the project by the deadline. The duration of this project will last from August 2021 to October 2021.

4 Schedule

4.1 Activity Dependencies and Schedule

PROJECT TITLE

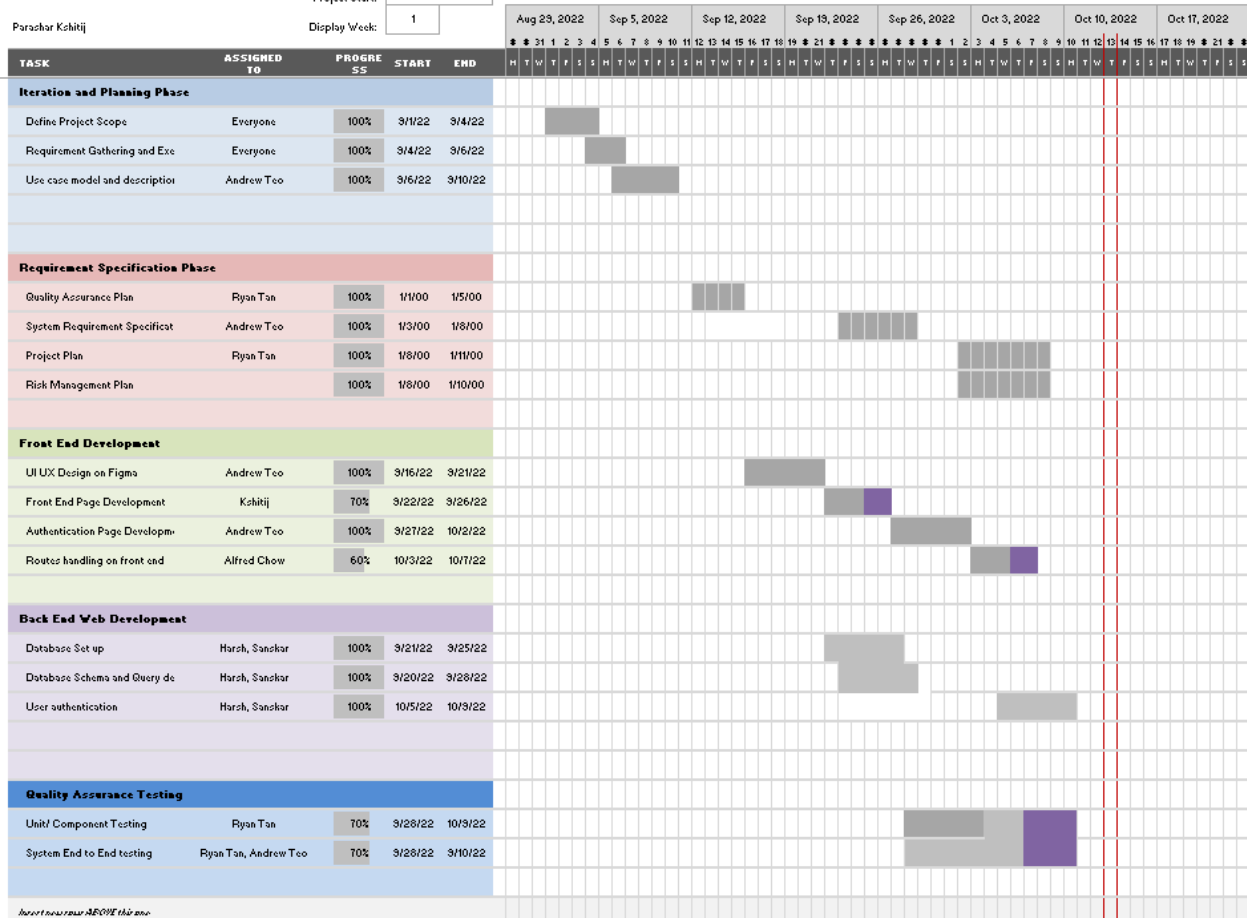
Company Name W/snnabFAANG
Project Lead

Parashar Kohliij

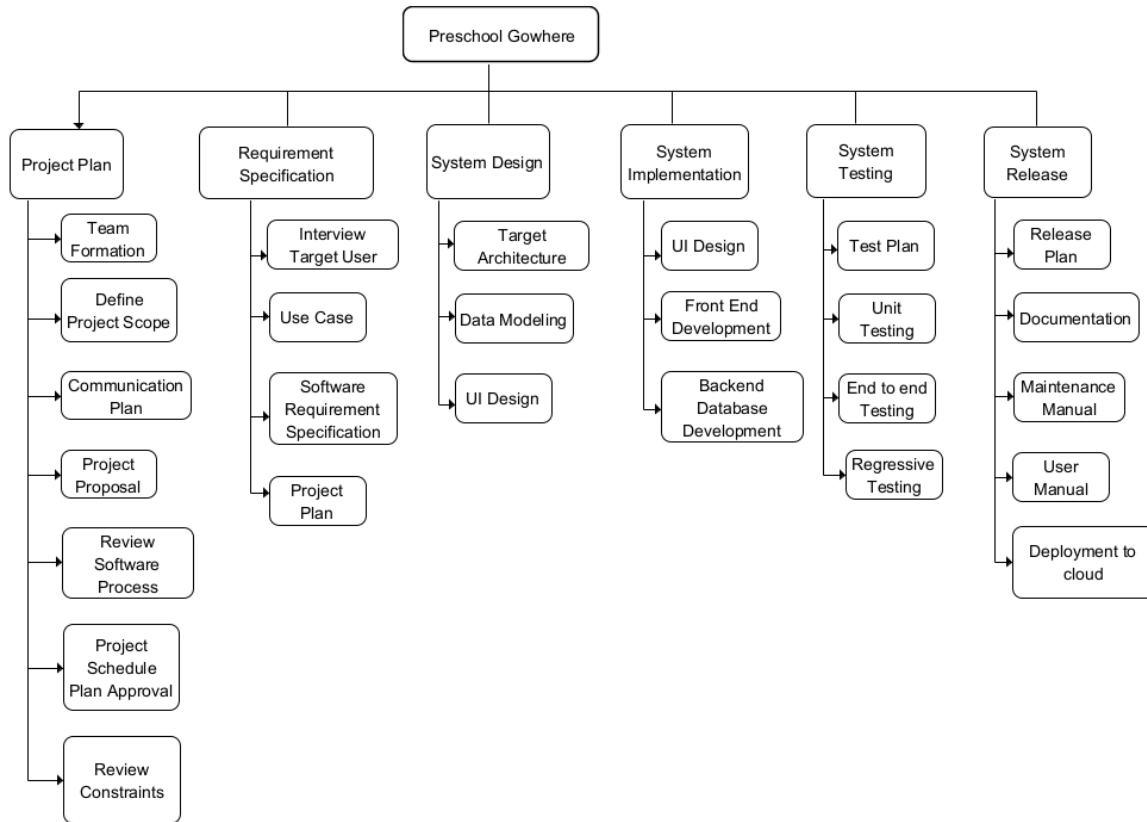
Project Start: Thu, 3/1/2022
Display Week: 1

SIMPLE GANTT CHART by Vertex42.com

<https://www.vertex42.com/ExcelTemplate/simple-gantt-chart.html>



4.2 Work breakdown Structure



4.3 Work Packages

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

1. Project Plan
2. Requirement Specification
3. User Interface
4. Technical Architecture
5. Data Modeling
6. Coding & Unit Testing
7. Integration and 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	10 days	–
X02	Requirement Specification	5 days	–
X03	User interface	6 days	–
X04	Technical Architecture	9 days	X01, X02, X03
X05	Data Modeling	7 days	X04
X06	Coding and Unit Testing	14 days	X05
X07	Integration and System Testing	14 days	X06

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

Preschool Gowhere supports the following proposed function:

User:

- Get information about specific preschool within their specified criteria
- Get email about contact details about specific preschool
- View all preschools in Singapore and any information relevant
- Bookmark any preschools of interest

Measurement of unadjusted function points is based off the five primary elements of those function:

- Inputs
- Outputs
- Logical Files
- Interfaces

Each element ranges from Low Complexity, Medium Complexity to High Complexity

Element	Complexity	Detail
Inputs	Low	Account Registration
	Medium	Customised Search Request
Outputs	Medium	Display all preschools in Singapore
	Medium	Displaying preschools from customised

		search
Inquiries	Low	Login
	Low	View bookmarked preschools
Logical Files	Low	Account Registration
	Low	Customised search
Interfaces	Low	Query of results from database
	Low	Update and modify bookmarked flats to databasew

Calculation of Unadjusted Function Points:

Characteristics	Low		Medium		High	
Inputs	1	x 3	1	x 4	0	x 6
Outputs	0	x 4	2	x 5	0	x 7
Inquiries	2	x 3	0	x 4	0	x 6
Logical Files	2	x 7	0	x 10	0	x 15
Interfaces	2	x 5	0	x 7	0	x 10
Unadjusted FP	23		14		0	
Total=L+M+H	37					

5.1.2 Adjusted Function Points

Influence Factors	Score	Detail
Data Communications	4	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	2	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	3	Some security or timing considerations are included.
Transaction Rate	5	Daily peak transaction 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	More than 30% of transactions are interactive data entry
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 use at the 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	38	
Influence Multiplier $= \text{Total score} * 0.01 + 0.65 = 38 * 0.01 + 0.65 = 1.03$		
Adjusted FP $= \text{Unadjusted FP} * \text{Influence Multiplier} = 37 * 1.03 = 38.11$		

Scoring (0-5)
0 = No influence
1 = Insignificant Influence
2 = Moderate Influence
4 = Significant Influence
5 = String influence

5.1.3 Lines of code

According to Quantitative Software Management, each Function Point requires 35 lines of code if the application is implemented using Javascript.

Therefore, we have: **Lines of Code** = $38.11 \text{ FP} \times 35 \text{ LOC/FP} = 1333 \text{ LOC}$

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 includes 5 days in a week
- Effort = Size / Producton Rate = (1333 LOC) / (39 LOC/ PD) = 34 PD
- Duration = $3 \times (\text{Effort})^{1/3} = 3 \times (34)^{1/3} = 9.7 \text{ D}$
- Initial Schedule = 9.7 Days / 5 days a week = 1.94 weeks
- Team size = 34 PD / 9.7 D = 3.5 P = 4 Persons
- Working hours include 8 hours in a working day
- Total person-hours (PH) = 34 PD x 8 hours = 272 PH

5.3 Cost Estimates

Hardware:

Developer workstations:

6 - Lenovo ThinkPad P16 Mobile Workstation	\$18000
12th Generation Intel i5-12699HX Processor	
8 GB RAM	
256GB SSD	

Software:

Software License Provided by Third Party

Microsoft Visual Studio Code	\$0.00
Supabase	\$0.00

The customer will supply the required hardware and software necessary to run the back-end and front-end web server. Team WannabeFAANG is not responsible in any way for supplying said systems.

Preschool GoWhere 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. Preschool GoWhere 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 Deliverables	Estimated Deadline
Requirements Specification Plan	Sept 29, 2022
Project Plan	Oct 13, 2022
Risk Management Plan	Oct 13, 2022
Module/System Test Plan	Oct 20, 2022
System Release (Demo of application)	Oct 27, 2022

7 Best Practice Checklist

Best Practices
<ul style="list-style-type: none">• To ensure accountability and future references, every procedure must be documented• All paperwork must follow a consistent and standardized structure.
<ul style="list-style-type: none">• Pay close attention to all requirements• Watch for thoroughness and accuracy, and steer clear of ambiguity.• Documentation must be clear and comprehensive in order for staff members to understand it when they refer to it in the future.• A thorough explanation of the functional requirements for the application should be included in the requirement documentations.
<ul style="list-style-type: none">• Ensure simplicity and reduce complexity when able to do so.• Minimize interfaces between modules, procedures and data to allow for loose coupling.• Aim to meet customer's requirements instead of making fancy or flashy features.
<ul style="list-style-type: none">• Allow for visibility (Able to see the builds of the product in order to measure current progress)• Team manager must have good communication with his employees/team• Developers must be required to submit their code for code review so that the team manager may monitor the employee's development and make sure she is heading in the right direction.
<ul style="list-style-type: none">• Allow flexibility and plan for continuous change.• Source code, manuals, tests all should have revision numbers and history comments• New revisions must be approved by either team manager or Quality assurance manager before being made. After being made, it should also be examined to see if the quality is acceptable.• Use a configuration management system and make processes
<ul style="list-style-type: none">• Plan ahead and do not under-estimate• To prevent delays in the process, it's critical to employ precise time, manpower, and meeting estimations.• Keep more time for integration, testing, maintenance and documentations as these processes are extremely crucial
<ul style="list-style-type: none">• Testing of software will include both black box testing and white box testing. It will first perform unit testing, followed by functional, integration and acceptance testing.

8 Risk Management

The following risks have been identified for PreSchool GoWhere project

Specification Delays

Impact severity : High

Probability: 10%

Zone:Yellow

Impacts: The PreSchool GoWhere project's schedule for all of the following phases will be pushed back if the specification is not finalized on time.

Risk Reduction: Monitor progress of specification carefully and minimize delays

More Changes to functional requirements than expected

Impact Severity: High

Probability: 15%

Zone : Yellow

Impacts: Changes to functional requirements may result in an update/change in the web application features. A redesign of the application might even be required, depending on how much modification is required.

Risk reduction: When describing the application's functional requirements, be thorough. Push out more prototypes to seek customer feedback before building the full feature of the application.

Finding customer feedback

Impact Severity: Medium

Probability: 15%

Impacts: Lack of client input might result in ambiguity over the project's suitability and appropriateness. For instance, if we do not get enough feedback from parents finding a preschool, we would not be able to have a clear idea on whether our application can efficiently help solve all their issues of finding a suitable preschool.

Risk reduction: Start early by making prototypes and getting feedback from the specific target audience of this project.

Staff/team members leaving before project complete

Impact Severity: Moderate (depending on current manpower allocation)

Probability: 20%

Impacts: More workload for remaining employees. This might result in more issues affecting the quality of product if we cannot find a substitute that is as competent as the employee that left.

Risk reduction: Ensure that team members agree to stay throughout the course of the project, this can be done via an agreement. Give out more incentives to help reduce the probability of them leaving the project halfway.

9 Quality Assurance

By conducting thorough software testing, the project will guarantee the necessary quality. The specific test procedures and details shall be provided in the System Test Plan. In this PreSchool GoWhere project, there will be two main testing methodologies, Unit Testing and Integration testing. Unit Testing involves testing system components individually to ensure that each individual component works well on its own. On the other hand, Integration tests verify that different modules or services used by your application work well together. Also, we will carry out acceptance testing once the product passes both the above tests. Acceptance tests are formal tests that verify if a system satisfies business and customer requirements. They require the entire application to be up running during testing. They can even assess the system's performance and, if specific objectives are not achieved, reject adjustments.

Furthermore, these methodologies will be used to test important aspects of PreSchool GoWhere.

- System Function will be tested to ensure that software flaws will be eliminated
- Algorithmic Function will be tested to ensure that heuristic aspects of the project (such as searching for the most suitable preschool) so as to provide a more accurate result for the end user.
- User Friendliness will be tested to ensure that PreSchool GoWhere users are able to easily and efficiently navigate through the app interfaces with little to no relearning/recap required, allowing for a natural user experience.

Project GoWhere makes broad use of realistic test cases to ensure that it can perform all the necessary functional requirements and efficiently solve the problems faced by parents finding difficulty in searching for a preschool for their children.

Software Quality consists of the following

- 1) Functionality - Application's functions cater to the customer's requirements and are implemented correctly.
- 2) Reliability - Application is able to perform under specific conditions.
- 3) Usability - users can easily use the application and understand the intended usage of various features in the application
- 4) Efficiency - good software principles and coding practices used to ensure efficiency
- 5) Maintainability - improvements and updates of code is easy to add. Bugs can be dealt easily
- 6) Flexibility - System can adapt quickly to changes

10 Monitoring and Control

In order to successfully track the development of a software project, many procedures will be needed. The following are some of the more significant procedures:

Identification of major risks associated with the project.

Early detection of risks aids in giving enough time to implement preventive actions, considerably reducing these hazards. By doing this, we can lessen the chance that the project will encounter further hazards as it is being developed.

Timeline Planning and task decomposition

An accurate timeline can be assembled by breaking down tasks into measurable subcomponents and estimating requirements for each task. This decomposition can also aid in task assignment and balancing. Throughout the implementation phase, these subcomponents allow for a better measurement of progress.

Regular reviews of project progress

Throughout the duration of the PreSchool GoWhere project, weekly team meetings will be held to review the status of all project tasks, including management, planning, analysis, development, and software testing. Additionally, the team manager must require developers to make their code available for code review so that he can track the current progress and ensure that the employee is on the right track. This is to ensure that the code is up to quality.

WANNABE FAANG

RISK MANAGEMENT PLAN

Version *1.1*
02/10/2022

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ryan Tan	28092022	Andrew Teo	28092022	Initial Risk Management Plan draft
1.1	Ryan Tan	01102022	Andrew Teo	02102022	Update Risk Management Plan

UP Template Version: 11/30/06

TABLE OF CONTENTS

1 INTRODUCTION	1
1.1 Purpose Of The Risk Management Plan	1
2 RISK MANAGEMENT PROCEDURE	1
2.1 Process	1
2.2 Risk Identification	1
2.3 Risk Analysis	1
2.3.1 Qualitative Risk Analysis	1
2.3.2 Quantitative Risk Analysis	2
2.4 Risk Response Planning	2
2.5 Risk Monitoring, Controlling, And Reporting	2
3 TOOLS AND PRACTICES	2
RISK MANAGEMENT PLAN APPROVAL	3
APPENDIX A: REFERENCES	4
APPENDIX B: KEY TERMS	5

1 INTRODUCTION

1.1 PURPOSE OF THE RISK MANAGEMENT PLAN

A risk is an event or condition that, if it occurs, could have a positive or negative effect on a project's objectives. Risk Management is the process of identifying, assessing, responding to, monitoring, and reporting risks. This Risk Management Plan defines how risks associated with the PreSchool GoWhere project will be identified, analyzed, and managed. It outlines how risk management activities will be performed, recorded, and monitored throughout the lifecycle of the project and provides templates and practices for recording and prioritizing risks.

The Risk Management Plan is created by the project manager in the Planning Phase of the CDC Unified Process and is monitored and updated throughout the project.

The intended audience of this document is the project team, project sponsor and management.

2 RISK MANAGEMENT PROCEDURE

2.1 PROCESS

The project manager, Parashar Kshitij, working with the project team and project sponsors will ensure that risks are actively identified, analyzed, and managed throughout the life of the project. Risks will be identified as early as possible in the project so as to minimize their impact. The steps for accomplishing this are outlined in the following sections. The QA manager, Andrew Teo, will serve as the Risk Manager for this project.

2.2 RISK IDENTIFICATION

Risk identification will involve the project team, appropriate stakeholders, and will include an evaluation of environmental factors, organizational culture and the project management plan including the project scope. Careful attention will be given to the project deliverables, assumptions, constraints, WBS, cost/effort estimates, resource plan, and other key project documents.

A Risk Management Log will be generated and updated as needed and will be stored electronically in the project library located at <file location>.

2.3 RISK ANALYSIS

All risks identified will be assessed to identify the range of possible project outcomes. Qualification will be used to determine which risks are the top risks to pursue and respond to and which risks can be ignored.

2.3.1 Qualitative Risk Analysis

The probability and impact of occurrence for each identified risk will be assessed by the project manager, with input from the project team using the following approach:

Probability

- High – Greater than 80% probability of occurrence
- Medium – Between 20% and 80% probability of occurrence
- Low – Below 20% probability of occurrence

Impact

Impact	High			
	Medium			
	Low			
		Low	Medium	High
Probability				

- **High** – Risk that has the potential to greatly impact project cost, project schedule or performance
- **Medium** – Risk that has the potential to slightly impact project cost, project schedule or performance
- **Low** – Risk that has relatively little impact on cost, schedule or performance

Risks that fall within the RED and YELLOW zones will have risk response planning which may include both a risk mitigation and a risk contingency plan.

2.3.2 Quantitative Risk Analysis

Analysis of risk events that have been prioritized using the qualitative risk analysis process and their affect on project activities will be estimated, a numerical rating applied to each risk based on this analysis, and then documented in this section of the risk management plan.

2.4 RISK RESPONSE PLANNING

Each major risk (those falling in the Red & Yellow zones) will be assigned to a project team member for monitoring purposes to ensure that the risk will not “fall through the cracks”.

For each major risk, one of the following approaches will be selected to address it:

- **Avoid** – eliminate the threat by eliminating the cause
- **Mitigate** – Identify ways to reduce the probability or the impact of the risk
- **Accept** – Nothing will be done
- **Transfer** – Make another party responsible for the risk (buy insurance, outsourcing, etc.)

For each risk that will be mitigated, the project team will identify ways to prevent the risk from occurring or reduce its impact or probability of occurring. This may include prototyping, adding tasks to the project schedule, adding resources, etc.

For each major risk that is to be mitigated or that is accepted, a course of action will be outlined for the event that the risk does materialize in order to minimize its impact.

2.5 RISK MONITORING, CONTROLLING, AND REPORTING

The level of risk on a project will be tracked, monitored and reported throughout the project lifecycle.

A “Top 10 Risk List” will be maintained by the project team and will be reported as a component of the project status reporting process for this project.

All project change requests will be analyzed for their possible impact to the project risks.

Management will be notified of important changes to risk status as a component to the Executive Project Status Report.

3 TOOLS AND PRACTICES

A Risk Log will be maintained by the project manager and will be reviewed as a standing agenda item for project team meetings.

RISK MANAGEMENT PLAN APPROVAL

The undersigned acknowledge they have reviewed the **Risk Management Plan** for the PreSchool GoWhere project. Changes to this Risk Management Plan will be coordinated with and approved by the undersigned or their designated representatives.

Signature: _____ Date: _____
Print Name: _____
Title: _____
Role: _____

Signature: _____ Date: _____
Print Name: _____
Title: _____
Role: _____

Signature: _____ Date: _____
Print Name: _____
Title: _____
Role: _____

Signature: _____ Date: _____
Print Name: _____
Title: _____
Role: _____

APPENDIX A: REFERENCES

The following table summarizes the documents referenced in this document.

Document Name and Version	Description	Location
Risk register	Log of all the risks associated with this project	https://docs.google.com/document/d/1dSfi0SDKY_ei9UpZxHmnZtDVRZuMhz_h7uR7cZahTUU/edit?usp=sharing

APPENDIX B: KEY TERMS

The following table provides definitions for terms relevant to the Risk Management Plan.

Term	Definition
Risk Log	A log file containing all the risks associated with this project.

WANNABE FAANG

**DESIGN REPORT ON SOFTWARE
MAINTAINABILITY**

Version 1.2

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Andrew Teo

Alfred Chow

Ryan Tan

Team WannabeFAANG

Document Change Record

Revision	Description of Change	Approved by	Date
1.0	Initial Template	Alfred Chow	15/10/22
1.1	Add Design Strategies, Architectural Design Patterns	Ryan Tan	17/10/22
1.2	Revised Edition	Andrew Teo	21/10/22

Table of Contents

1. Design Strategies	4
1.1. The Planning Phase Before Development	4
1.2. The Process of Developing	4
1.3. Correction by Nature	5
1.4. Enhancement by Nature	5
1.5. Maintainability Practices	5
2. Architectural Design Patterns	6
3. Software Configuration Management Tools	7
3.1. MediaWiki	7
3.2. Github	7
3.3. Google Drive	7
3.4. Trello	7

1. Design Strategies

1.1. The Planning Phase Before Development

We anticipated and planned for various continuous enhancements that we would need to do after releasing this application from the beginning. As it receives feedback from various users, PreSchool GoWhere would need to continuously update its features along the way. Improvements can be in the form of effectiveness of our features, user friendliness or even making the user experience and navigation a more natural one. Moreover, once the number of users for our application increases, we would also need to scale the application to cater to higher traffic rates in order for the application to be reliable for all users. As such, maintainability and scalability would be extremely important for us to ensure that we would be able to easily maintain or update our application.

Hence, we have decided to adopt the Model View Controller (MVC) architecture to help ensure maintainability of our code and a robust design architecture for our application. MVC pattern adopts a separation between the software's business logic and display, hence allowing loose coupling. This provides for a better division of labor and improved maintenance of code if we were to update the app features every now and then. It allows ease in changing and implementing new functions without breaking and affecting the existing functions. Also, since the view layer is separated in this MVC architecture, it is easier to update the user interface components of the application.

1.2. The Process of Developing

We will be testing our software application in a small and test-driven environment to allow for easier debugging and more efficient means of testing. Fewer bugs and errors are the primary benefit of test driven development, hence saving more time in our process of developing and testing the application. For this PreSchool GoWhere project, we will be focusing on parents who may be finding a preschool for their kids. We will release survey forms both online and in selected locations such as near preschools, then ask them to try out our software prototype. They will perform the role of the tester and provide valuable feedback on the design and usability of the application. After testing, PreSchool GoWhere will host this application and advertise it in Singapore to cater to more parents all over Singapore.

1.3. Correction by Nature

We will update and continuously correct our application when testing it.

1.3.1 Corrective Maintainability - Fault detection carried out by testing

1.3.2 Preventive Maintainability - Features are implemented atomically, each feature is individually tested, and errors are quickly found.

1.4. Enhancement by Nature

While testing the program, we will make improvements and the following should be aimed to achieve in order to ensure the success of the project.

1.4.1 Adaptive Maintainability - Application should be relatively easy to modify for a different operational context.

1.4.2 Perfective Maintainability - After the product is being delivered, quickly spot any issues or problems, and fix it to cut down on maintenance expenses and time.

1.5 Maintainability Practices

Over the course of our project, we have established the following maintainability policies to retain quality in both process and product. We ensure our code to be readable, standardized our documentations and delivered proper version control.

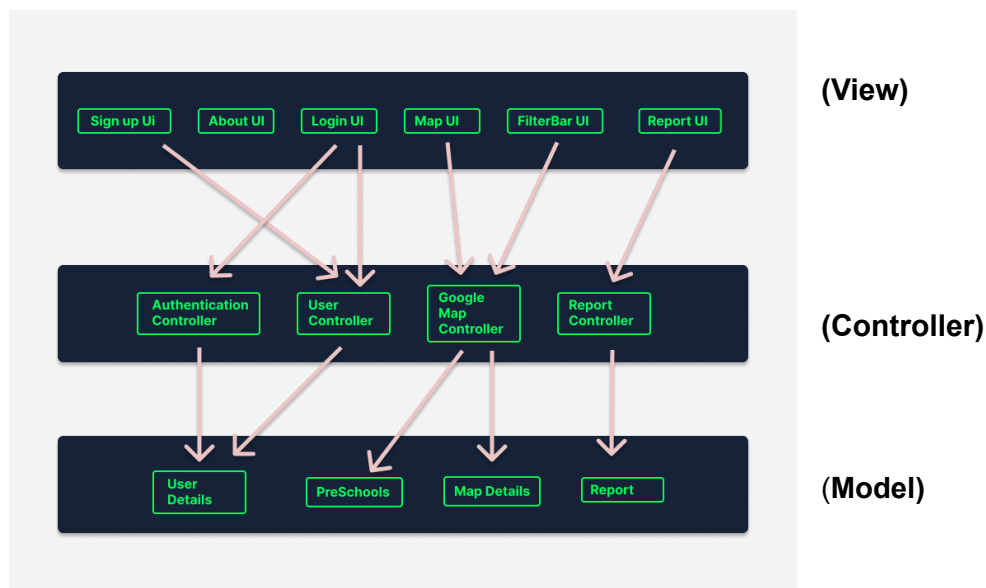
2. Architectural Design Patterns

PreSchool GoWhere employs a client-server architecture, a distributed application framework that divides work between service requesters, also known as clients, and resource or service providers, known as servers. The client makes up the website interface that interacts dynamically with the user. The client would connect to the backend server that holds our SQL database. For instance, this enables our client to send GET requests to the backend server which would retrieve the data needed and send it to the client side. The website interface can then dynamically update itself to present the updated results or data to the user.

PreSchool GoWhere's interface uses the Model-View-Controller (MVC) architectural design pattern. The model layer is where the persistent user data is held and records the usage from the user interface level. The view layer shows the components to display data from the model layer. In our case, we are using ReactJS components to dynamically display the updated data from the model layer. The controller layer holds the application logic and receives/carries out commands from the user to update the view or model layer.

Our software application adopts the Model View Controller (MVC) architecture to help ensure maintainability of our code and a robust design architecture for our application. MVC pattern adopts a separation between the software's business logic and display, hence allowing loose coupling but high cohesion. This provides for a better division of labor and improved maintenance of code if we were to update the app features every now and then.

Design of our mvc architecture:



3. Software Configuration Management Tools

This is where we will be discussing the version control management, and how we track our progress and updates.

3.1. MediaWiki

Being a free and open-source application, MediaWiki is utilized since it is simple for beginners to learn. There are several FAQs available that can instruct users on the functions they need. Users can produce their information using a variety of functionalities and in a variety of ways. Additionally, it enables simultaneous user editing of the page. Therefore, changing the page won't cause any information to be lost. It is used as it is easy for beginners to pick up.

3.2. Github

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub is chosen for its familiarity and support provided by various IDE applications. Additionally, GitHub supports issue tracking. Whether it's a software bug, code enhancement or documentation, users can open an issue, label them appropriately and assign them for other team members to resolve. All users will receive regular updates on the project's status. Also, we also use trello to display the list of backlog that we have and notify one another on the implemented features that we have done.

3.3. Google Drive

Google Drive is employed for file storage as well as document backup. Users can easily share and store files using this service within the team. Moreover, they can also concurrently edit documents together using this service. This service serves as a great means for team members to store their documentations and work together on the same project files.

3.4. Trello

Trello is a visual tool that enables us to have a bird's eye view on all the todos and backlogs of a project. It allows us to easily communicate about specific issues with regards to development of software. Hence, this service helps team members work more efficiently with a better means of communication.

PreSchool GoWhere

Change Management Plan

Team WannabeFAANG

Version 1.2

Date: 15/11/2022

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ryan Tan	01/10/2022	Andrew Teo	01/10/2022	Initial Change Management Plan
1.1	Ryan Tan	14/10/2022	Alfred Chow	14/10/2022	Content Addition
1.2	Ryan Tan	15/10/2022	Andrew Teo	14/10/2022	Final Version

VERSION HISTORY	2
1 Introduction	4
1.1 Purpose of The Change Management Plan	4
2 Change management Process	4
2.1 Change Request Process Flow Requirements	4
2.2 Change Request Form and Change Management Log	5
2.3 Evaluating and Authorizing Change Requests	6
2.3.1 Change Control Board	8
3 Responsibilities	10
Appendix A: Change Management Plan Approval	13
Appendix B: References	14
Appendix C: Key Terms	15
Appendix D: Change Request Form Example	16
Appendix E: Change Management Log Template	16

1 Introduction

1.1 Purpose of The Change Management Plan

The Change Management Plan documents and tracks the necessary information required to effectively manage project change from project inception to delivery.

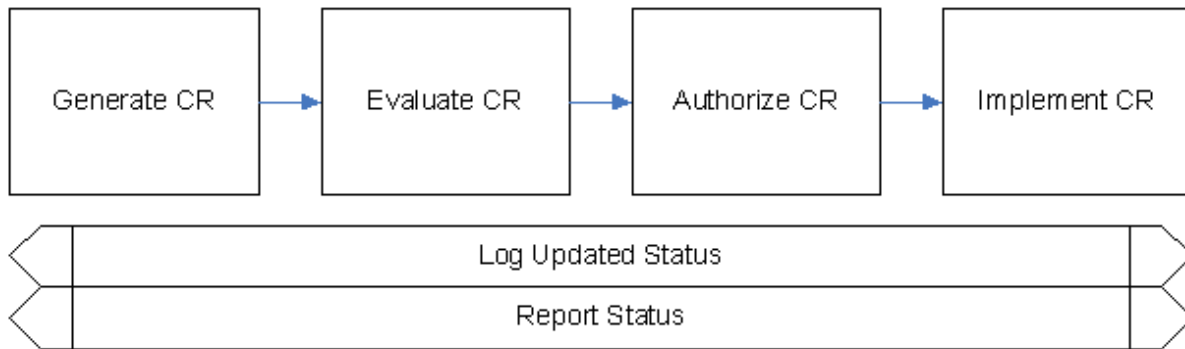
The Change Management Plan is created during the Planning Phase of the project. Its intended audience is the project manager, project team, project sponsor and any senior leaders whose support is needed to carry out the plan.

2 Change management Process

The Change Management process establishes an orderly and effective procedure for tracking the submission, coordination, review, evaluation, categorization, and approval for release of all changes to the project's baselines.

2.1 Change Request Process Flow Requirements

Step	Description
Generate CR	A submitter completes a CR Form and sends the completed form to the Change Manager
Log CR status	The Change Manager enters the CR into the CR Log. The CR's status is updated throughout the CR process as needed
Evaluate CR	Project personnel review the CR and provide an estimated level of effort to process, and develop a proposed solution for the suggested change
Implement	If approved, make the necessary adjustments to carry out the requested change and communicate CR status to the submitter and other stakeholders



2.2 Change Request Form and Change Management Log

Element	Description
Date	The date the CR was created
CR#	Assigned by the Change Manager
Title	A brief description of the change request
Description	Description of the desired change, the impact, or the benefits of a change should also be described
Submitter	Name of the person completing the CR Form and who can answer questions regarding the suggested change
Phone	Phone number of the submitter
E-Mail	Email of the submitter
Product	The product that the suggested change is for
Version	The product version that the suggested change is for
Components Affected	The components that would be affected temporarily/ permanently if the change were to be implemented
Priority	A code that provides a recommended categorization of the urgency of the requested change (High, Medium, Low)
Conditions	Any condition if change were to take place

2.3 Evaluating and Authorizing Change Requests

Change requests are evaluated using the following priority criteria:

Priority	Description
Critical	The impact of the change is critical and is able to: <ol style="list-style-type: none">1. Affect the basic functionalities of the product. (E.g. crashing of the website, database downtime)2. Result in loss of sensitive data and information.3. Affect all users and multiple departments
High	The impact of the change is able to : <ol style="list-style-type: none">1. Impact important business processes/ systems and may result in an outage.2. Affect major users
Medium	<p>Change request involves a change which approval is mandatory as specified by the client.</p> <p>Such change requests has little/ no impact to the performance and stability of the product.</p> <p>Due to its nature, the change request has to be reviewed independently to evaluate the necessity for the change.</p>
Low	<p>Change request involves a change that impacts the website design aesthetics, fixture of minor user experience bugs or additional features to be added as per requested by client.</p> <p>Such change requests do not impact the overall performance and stability of the system, and can be rejected if unneeded. They should be reviewed accordingly.</p>
Insignificant	Change request may contain incomplete or inaccurate information, and the issue may be non-existent. Else, the change adds insignificant improvements to the existing product.
Mandatory	Additional features requested by the client that have to be included. The product's performance and stability remains unchanged.

Change requests are evaluated and assigned one or more of the following change types:

Type	Description
Scope	Change affecting scope
Time	Change affecting time
Duration	Change affecting duration
Cost	Change affecting cost
Resources	Change affecting resources
Deliverables	Change affecting deliverables
Product	Change affecting product
Processes	Change affecting process
Quality	Change affecting quality
Reliability	Change affecting stability of the system
Security	Change affecting the vulnerability of the system towards exploits and security attacks.

Change requests are evaluated and assigned one of the following status types:

Status	Description
Open	Entered/Open but not yet approved or assigned
Work in Progress	CR approved, assigned, and work is progressing
In Review	CR work is completed and in final review prior to testing
Testing	CR work has been reviewed and is being tested
Approved	CR work has been reviewed by Project Manager/ Lead developer and has been approved for merge
Merged	CR work has been merged into main branch by DevOps personnel
Closed	CR work is complete, has passed all tests, and updates have been released
Rejected	CR work has been rejected by the approving entity

2.3.1 Change Control Board

Role	Name	Contact	Description
Project Manager	Parashar Kshitij		<ul style="list-style-type: none">• Oversees project status and progress• Approval and execution of project plan and additional features to be added• Assigns task and reports status of project to team members• Manage and motivate team• Represent team in communication with relevant stakeholders

Lead Developer	Malavade Sanskar Deepak		<ul style="list-style-type: none"> • Overall technical lead • Designing of technology stack to be used for project • Review and merge pull request made by team members • Reports to project manager development team and progress • Security of software product
QA Manager	Andrew Teo		<ul style="list-style-type: none"> • Oversee the overall product and process quality • Ensures that all functional and non-functional requirements are met • Record, analyze and distribute statistical information • Supervise QA engineer • Review test cases and plans developed by QA Engineer
Risk Manager	Ryan Tan		<ul style="list-style-type: none"> • Conduct weekly risk assessment meetings with the risk management team • Improvement or creation of risk

			<p>management plans to reduce and eliminate potential risk involved in project</p> <ul style="list-style-type: none"> • Ensure that these plans are executed
--	--	--	---

3 Responsibilities

[Provide a brief description of persons responsible for each step of the change management process for the project.]

Role	Name	Contact	Description
Project Manager	Parashar Kshitij		<ul style="list-style-type: none"> • Oversees project status and progress • Approval and execution of project plan and additional features to be added • Assigns task and reports status of project to team members • Manage and motivate team • Represent team in communication with relevant

			stakeholders
Lead Developer	Malavade Sanskar Deepak		<ul style="list-style-type: none"> • Overall technical lead • Designing of technology stack to be used for project • Review and merge pull request made by team members • Reports to project manager development team and progress • Security of software product
Change Manager	Alfred Chow		<ul style="list-style-type: none"> • Consistent communication with Project Manager about changes pertaining to the project • Oversees any form of changes made to the project • Analyse the impact of change towards the system • Evaluation

			and approval of change proposal
--	--	--	---------------------------------------

Appendix A: Change Management Plan Approval

The undersigned acknowledge they have reviewed the Preschool GoWhere **Change Management Plan** and agree with the approach it presents. Changes to this **Change Management Plan** will be coordinated with and approved by the undersigned or their designated representatives.

Signature: _____ Date: _____

Print Name: _____

Title: _____

Role: _____

Signature: _____ Date: _____

Print Name: _____

Title: _____

Role: _____

Signature: _____ Date: _____

Print Name: _____

Title: _____

Role: _____

Appendix B: References

The following table summarizes the documents referenced in this document.

Document Name and Version	Description	Location
PreSchool_GoWhere_Change_Management_Plan v1.2.doc	PreSchool GoWhere Change Management Plan Document	Wikimedia

Appendix C: Key Terms

The following table provides definitions for terms relevant to this document.

Term	Definition
Change Requests (CR)	<p>A change proposal that will be able to impact the project.</p> <p>Change requests can be submit internally within the project team, or externally if the project is open sourced on platforms such as Github.</p> <p>The Change Manager will be involved in evaluating and approving the change requests made.</p>
Change Request Board (CCB)	<p>Committee responsible for recommending and making decisions on requested changes for baselined work.</p>

Appendix D: Change Request Form Example

The example Change Request Form attached below can be used to submit changes during the life of the project.

Appendix E: Change Management Log Template

The detailed Change Management Log template attached below can be used to track and manage requested changes during the life of the project.

Preschool GoWhere ChangeLog

Project Name:	PreSchool GoWhere
Project Manager:	Parashar Kshitij

[illegible]

WANNABE FAANG

CONFIGURATION MANAGEMENT PLAN

Version 1.2

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Andrew Teo

Alfred Chow

Ryan Tan

Team WannabeFAANG

REVISION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ryan Tan	01/10/2022	Andrew Teo	01/10/2022	Initial Change Management Plan
1.1	Ryan Tan	14/10/2022	Alfred Chow	14/10/2022	Content Addition
1.2	Ryan Tan	15/10/2022	Andrew Teo	14/10/2022	Final Version

Table of Contents

REVISION HISTORY	2
Table of Contents	3
1 Introduction	4
1.1 Document Overview	4
1.2 Abbreviations and Glossary	4
1.3 References	4
1.3.1 Project References	4
1.4 Conventions	5
2.Organisation	5
2.1 Activities and responsibilities	5
3.Configuration Identification	8
3.1 Identification rules of configuration items	8
3.1.1.1 Identification of a configuration item	8
3.1.1.2 Version number of a configuration item	9
3.1.2 Identification rules of documents	10
3.1.2.1 Description of documents identifiers.	10
3.1.2.2 Definition and evolution of the revision index	10
3.2 Reference configuration identification	10
3.3 Configuration Baseline Management	11
4 Configuration control	12
4.1 Change Management	12
5 Configuration support activities	12
5.1 Configuration Status Accounting	12
5.1.2 Setting up configuration status	13
5.1.3 Configurations status diffusion	13
5.1.4 Configuration status records storage	13
5.2 Configuration audits	13
5.4 Configuration management plan maintenance	14

1 Introduction

1.1 Document Overview

This document contains the software configuration plan of PreSchool GoWhere

1.2 Abbreviations and Glossary

Abbreviations

.SCM : Software Configuration Manager

.SVN: Subversion, which will be used to manage and track changes made to the repository

.SRS: Software requirement specification

.VDD: Version description document

Glossary

. Branch : a line of development that exists independently of another line, but can be merged together in the future

. Version: state of configuration at a specific point in time.

. Variant: versions that coexist, but are different from other versions.

1.3 References

1.3.1 Project References

#	Document Identifier	Document Title	Issuance Date
Ref 1	PG-05	Project Plan	12/10/2022

Ref 2	PG-02	Use Case Model	07/09/2022
Ref 3	PG-03	System Requirement Specification	21/09/2022
Ref 4	PG-04	Quality Plan	21/09/2022

1.4 Conventions

Conventions used in project:

- Typographical Conventions
- Coding Conventions

2. Organisation

The software configuration is managed by members of the project, with specific tools.

Responsibilities are shared between

- The software configuration manager: Andrew Teo
- Project Manager: Parashar Kshitij
- Technical Manager: Ryan Tan

2.1 Activities and responsibilities

The various activities required to manage the software configuration and the person responsible for the functions are listed as below:

Activities when setting up the project	Person responsible
Identify items to be configured	SCM
Install the bug repository tool	SCM
Install the software configuration repository tool	SCM
Structure and manage the reference space	SCM
Determine the configuration process	SCM

Activities during the project lifecycle	Person Responsible
Export components for modification, testing and delivery	SCM
Set in control validated components	SCM
Create version, write version delivery document	SCM
Approve the reference configurations	Project Manager
Verify version to be delivered and authorise deliveries	Project Manager
Backup spaces	SCM
Do configuration audits	Quality Manager

Management activities	Person responsible
Manager versions and archives	SCM
Manage configuration records	SCM
Produce reports and statistics	SCM

At the end of an activity of the project:

Activities	Person Responsible
Do a configuration freeze	SCM
Present the configuration state of components impact by the activity made	SCM
Present the documentation state of components impacted by the activity made	SCM

During the audit of a configuration management process:

Activities	Person Responsible
Do the configuration management process audit	Project Manager
Present the records of the configuration management process	SCM

Present the quality records of the configuration management process	Quality Manager
Present the records of the documentation management process	SCM

3.Configuration Identification

3.1 Identification rules of configuration items

3.1.1.1 Identification of a configuration item

- Code
 - Source Code
- Documentation
 - Project proposal
 - System requirement specification
 - Quality Plan
 - Project Plan
 - Risk Management Plan
 - Test Plan
- System Designs
 - Use case models
 - Conceptual models
 - Software architecture
 - Software prototype
 - Unit test
 - Coverage test
- Specification
 - Project requirements specification
 - Unit test

3.1.1.2 Version number of a configuration item

The attribution of a version number is a prerequisite to any delivery of any configuration item. This number shall be incremented before a new delivery, if the product or its documentation were modified.

The definition rules of a version number are the following:

- The version number is of the form MAJOR.MINOR.PATCH. The version must be incremented when there's any changes
- Major release:
 - Significantly huge change to current business logic of the system
 - Change in technology stack used to operate the current system
 - Addition or removal of crucial functions which are able to impact the operation of the system
 - Significant redesign of user interface
- Minor Release
 - Small improvements and bug fixes
 - Changes do not affect the user interface and experience
- PATCH:
 - Compatibility issues
 - Grammatical or spelling error issues
 - Optimisation for improvement of performance

3.1.2 Identification rules of documents

3.1.2.1 Description of documents identifiers.

The project adopts a standard naming convention for documents:

XXX_<document type>_<document number>_<revision index>_<date>

where:

- “**Document type**” is the type of document
- “**Document number**” is an incremental number specific to the document type. Each document type will have an incremental number tagged to it.
- “**Revision index**” designates the approved iteration of the document. E.g. V1 represents the first approved iteration, V2 the second etc
- “**Date**” is the date at which the document has been either created or modified. It must be updated every time the document is updated.

3.1.2.2 Definition and evolution of the revision index

The attribution of a revision index is a prerequisite to any delivery of a document or file. This index shall be incremented before the diffusion of a modified document.

3.2 Reference configuration identification

Each reference configuration is defined by:

- An identifier
- Its content listed in the corresponding version Delivery Description document
- The acceptance or validation reviews associated with the building of the reference configuration.

A reference configurations established for each design review and each test review of the project.

3.3 Configuration Baseline Management

The managed baselines are:

- Functional baseline (FBL), which describes the system functional characteristics;
- Allocated baseline (ABL), which describes the design of the functional and interface characteristics
- Product baseline (PBL), which consists of completed and accepted system components and documentation that identifies this products.

4 Configuration control

4.1 Change Management

The process for controlling changes to the baselines and for tracking the implementation of those changes are shown below.

Problem resolution:

- Changes requests are emitted from by the project manager according to the problem resolution process
- When a change request is accepted by the project manager, a branch is created in Github
- The branch identification format is as follows
 - <problem>_v<version number>
 - Problem: title of the problem
 - Version Number: the version of the branch (to track which branch contains the latest code)
 - Branch content contains the applied changes, which can be viewed by viewing the commit history details.

5 Configuration support activities

5.1 Configuration Status Accounting

Configuration Status Accounting (CSA) is the process to record, store, maintain and report the status of configuration items during the software lifecycle. All software and related documentation should be tracked throughout the software life.

5.1.2 Setting up configuration status

The SCM sets up the state of all versions and each configuration article with:

- The label
- The version number
- The creation date of the VDD

The SCM writes the VDD

5.1.3 Configurations status diffusion

The SCM and the quality manager write the VDD

5.1.4 Configuration status records storage

The records are stored in a configuration folder, which contains:

- The requests sorted by record number
- The software documents
- The VDD
- The configuration states sorted chronologically

5.2 Configuration audits

- Baseline audit
- Functional configuration audit
- Software configuration audit
- Penetration Testing audit

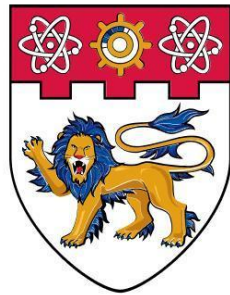
5.3 Reviews

Technical reviews during the project are related to the establishment of baselines and branches. SCM plays an important role in these reviews as SCM needs to help the teams to maintain consistency and control over what is produced.

- After establishment of the baseline, changes to the SCI can only be made under a formal change control procedure.
- Configuration reviews will be carried out periodically to verify the correctness of the configuration status
- Purpose of the configuration review is to make sure all the changes are recorded and the all project components are correctly identified, and document change has been noted.

5.4 Configuration management plan maintenance

The maintenance of the configuration management plan is under the responsibility of the QA Manager, who will have to update the CM plan and verify the correctness of the components developed. This will be conducted on a bi-monthly basis.



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

PreSchool GoWhere

Software Quality Assurance (SQA) Plan

Version 1.2

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Andrew Teo

Alfred Chow

Ryan Tan

Team WannabeFAANG

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Ryan Tan	01/10/2022	Andrew Teo	01/10/2022	Initial Change Management Plan
1.1	Ryan Tan	14/10/2022	Alfred Chow	14/10/2022	Content Addition
1.2	Ryan Tan	15/10/2022	Andrew Teo	14/10/2022	Final Version

Approvals

Submitting Organisation's Approving Authority:

Signature	Printed Name	Date	Phone Number	Position Title
	Parashar Kshitij	27/10/2022		Project Manager

CMS's Approving Authority

Signature	Printed Name	Date	Phone Number	Position Title

Revision History

Version	Date	Author	Description of Changes	Approved By
1.0	15/10/2022	Ryan Tan	Baseline Version	Andrew Teo
1.1	27/10/2022	Ryan Tan	Final Version	Andrew Teo

Table Of Contents

1. Introduction	1
2. Referenced Documents	1
3. Overview	2
4. Assumptions and Constraints	3
4.1 Assumptions	3
4.2 Constraints	3
4.2.1 Time	3
4.2.2 Budget	3
4.3 Risks	3
5. Release Approach	6
5.1 Rationale	6
5.2 Release Strategy	6
5.2.1 Release content	6
5.2.2 Release Schedule	7
5.2.3 Release Impacts	8
6. GLOSSARY	9
7. ACRONYMS	9
8 APPENDICES	9

PreSchool GoWhere

List of Figures

Figure 1: System Context Diagram	2
----------------------------------	---

List of Tables

Table 1: Referenced Documents	1
-------------------------------	---

1. Introduction

The following release plan contains detailed information pertaining to each release version of project PreSchool GoWhere. It also discusses the methodology and approach adopted by team WannabeFAANG in the process of Release Management, as well as assumptions and constraints faced by the team in designing a release management plan.

The intended audience for this Release Plan will be the entire project team, as well as any other relevant stakeholders who wish to have an overview and understanding of the release management operations by Team WannabeFAANG on project PreSchool GoWhere. The development of each release package is handled by the development team, which will then be handed over to the QA team for quality management of the release. The project manager will be critical in overseeing the release development as well as providing the final approval for the release distribution.

This release plan will also be a useful source of reference to the ecosystem of open source developers should the team intend to open source this project in the future.

2. Referenced Documents

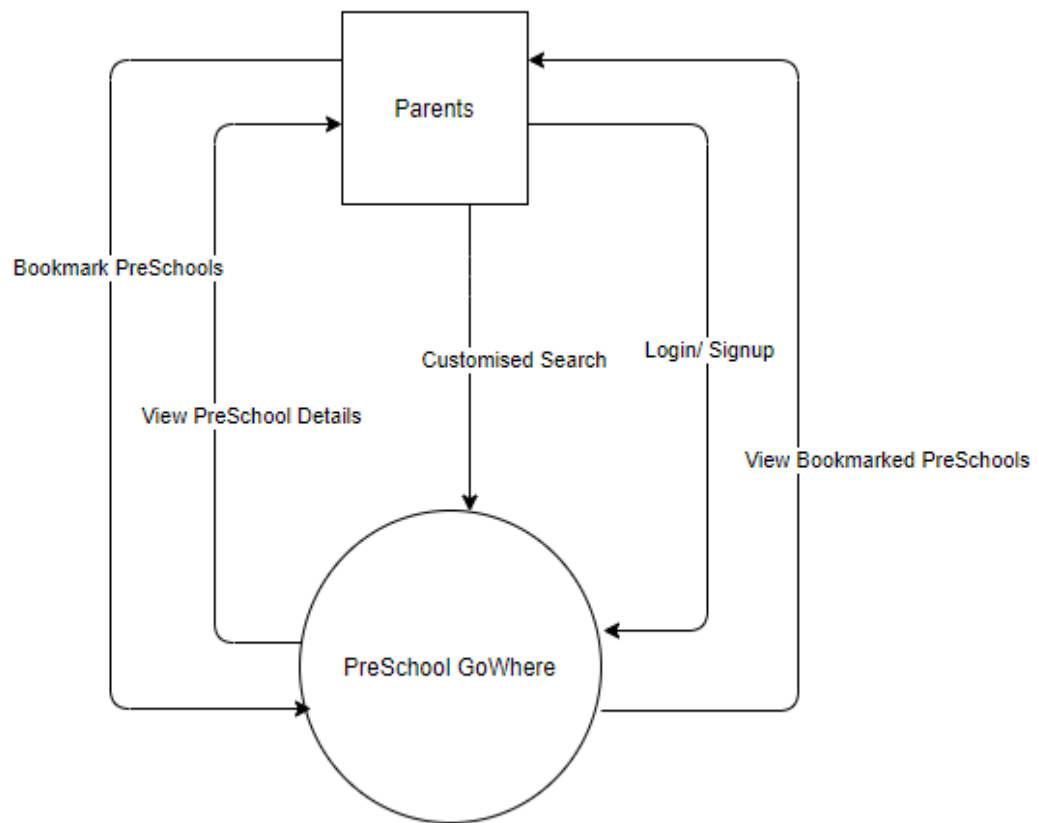
Table 1: Referenced Documents

#	Document Name	Document Number	Issuance Date
Ref 1	Use case model	PG-02	12/10/2022
Ref 2	System Requirement Specification	PG-03	21/09/2022

3. Overview

The figure below is a System Context Diagram which shows a high level overview of the operations behind PreSchool GoWhere.

Figure 1: System Context Diagram



4. Assumptions and Constraints

4.1 Assumptions

- The manpower size of project PreSchool GoWhere will remain constant throughout the entire course of the project, and there will be no change in the roles of team members involved in the project.
- Members will uphold their individual responsibility to complete each task delegated to them.
- There will be no changes in the pre-calculated cost and expenses that will be incurred during the course of the project.
- There will be no changes to the planned design and architecture to achieve the final product.

4.2 Constraints

4.2.1 Time

The timeframe given to complete the project and produce a working product is between 8-9 weeks, with all use cases mentioned in Use Case model to be developed and functioning.

4.2.2 Budget

A fixed budget is allocated for the project, no additional funding will be given during the course of the product development. As such, the project team will have completed the project by working around with the given budget and resources.

4.3 Risks

Table 2: Risks and Approach

Risk Type	Risks	Approach
Technical	Poor code quality (E.g. poor code readability, tightly coupled code). Impact: Can lead to poor software	Practice good coding conventions as well as adhere to good software development practices such as loose coupling. Peer review of codes can

	maintainability in the future, increasing maintenance cost.	also be conducted internally within the team or externally.
	<p>Introduction of additional features demanded by project sponsors. These features were not decided upon during the planning phase of the project.</p> <p>Impact:</p> <p>Addition of new features can distract team members from completing core features specified in the use case model.</p> <p>Threatens the time needed to complete the project</p>	<p>Effective communication is required between project manager and project sponsors to mitigate the addition of new features as much as possible.</p> <p>However, in the event where it is a necessity for the feature to be added, the project manager has to ensure that effort required to implement the feature does not affect the project completion deadline.</p>
	<p>Insufficient client engagement as well as client feedback.</p> <p>Impact:</p> <p>Creation of redundant features.</p> <p>Features may have bugs that escape the QA tests.</p> <p>Unable to determine if features will find a proper product market fit</p>	<p>Clients should be involved and integrated into the software development lifecycle.</p> <p>Client feedback will be gathered whenever there are new developments or updates.</p>
Management	<p>Creating too many features to be implemented during the project planning phase without considering factors such as project deadline.</p> <p>Impact:</p> <p>Can possibly threaten the project completion deadline.</p> <p>Drains the team resources fast.</p>	<p>Communication between team members is essential during the planning phase of the project.</p> <p>During the initial planning phase of the project, realistic expectations should be set, taking into consideration constraints</p>

		such as time and cost of running the project.
	<p>Team members are not committed to the project, and overall team productivity is low.</p> <p>Impact:</p> <p>Can lead to team members scrambling to complete the work near the project's deadline.</p> <p>Insufficient QA testing for products.</p> <p>Less time to cover possible edge cases, introduction of possible bugs in code.</p>	<p>Project Manager will need to ensure that the team stays up on task, and fulfill the deliverables required at the end of each sprint cycle. Any possible backlog incurred during the end of the sprint cycle should be cleared on a timely basis, instead of trying to clear the backlog towards the deadline.</p>

5. Release Approach

5.1 Rationale

PreSchool GoWhere adopts the Agile Methodology which heavily involves incremental development throughout the software development lifecycle, alongside multiple releases of the product. A stable baseline built (Version 1.0.0) will be released after all implementation and testing has been conducted and cleared from the backlog.

Any reported or discovered bugs, issues or feedback either from end-users or internally within the team will be reviewed by the team. End-users or team members can also suggest additional features that they would like to see in the product, which will also be reviewed by the team. Changes and updates will be released in subsequent versions if they have been approved, and can be released either in the form of a patch or a major release.

5.2 Release Strategy

The team has scheduled the first release of the product to be deployed on the 9th week since the debut of the project. All functionalities that have been specified in the SRS will be implemented. For the subsequent releases, they will be in the form of patch or minor release, which would contain minor bug fixes, as well as any new features requested by the user.

For major releases, they require a minimum period of 4 months to be completed. Such changes often involve a major overhaul to the system.

5.2.1 Release content

The table below shows each of the different release versions and the changes that entail them.

Table 3.1: Release Content - Initial Release

Release Version 1.0.0	
Functionality	Description
Authentication System	Fully implemented as defined in the SRS
Cloud hosted Database System	Fully implemented as defined in the SRS

Interactive Front End Webpage <ul style="list-style-type: none"> - Mobile compatible - Login Page - Signup Page - View Preschools page - Map display - Bookmarking of preschools 	Fully implemented as defined in the SRS
Backend Features: <ul style="list-style-type: none"> - Preschool Contact details email to user 	Fully implemented as defined in the SRS

Table 3.2: Release Content - Minor Release

Release Version 1.1.0 (Minor Release)	
Functionality	Description
Database System	Bug fixes, minor improvements
Front End page design	Bug fixes, minor improvements

Table 3.3: Release Content - Major Release

Release Version 2.0.0 (Major Release)	
Functionality	Description
System front end	Introduction of new major features or system overhaul
System backend	Introduction of new major features or system overhaul

5.2.2 Release Schedule

The project team expects to deploy the first release (version 1.0.0) of PreSchool GoWhere in the 9th week since the debut of the project. The first release will contain implementations of all core functionalities as mentioned in the SRS.

Subsequent minor releases will be released on a monthly basis, which contains bug fixes and minor improvements, as well as any functionalities that were requested by users after the first release.

Major releases will be done minimally every 3 months, which either contains an addition of a major feature to the system or a complete system overhaul.

5.2.3 Release Impacts

For each new release, the development team is required to make changes to the system, this presents a possibility of the incoming changes to affect existing features of the system which will then need to be respectively addressed. This will increase the amount of workload for the team. Any modifications to the system functionalities might also affect future developments to the system. It is imperative for the project manager to actively oversee the operations of the new release and to minimise any possible conflicts that may arise from the release.

With each new release of the system, there will also be a downtime of the system where the entire system has to be taken offline to commence the deployment of the system. During this downtime, users will not be able to access the system. This impact can be mitigated by scheduling the deployment of the release where traffic to the website is at its lowest. To inform users that the website is undergoing maintenance, the team can consider creating a placeholder website that notifies users that the system is undergoing maintenance.

5.2.4 Release Notification

Respective stakeholders will be notified after a release version is generated. The following table describes how the relevant stakeholders will be notified of the release version.

Table 4: Release Notification

Stakeholders	Notification Method	Information included in the notification	Timeframes for Receipt of Notification
Users	Emails, System Notification	Change Log	Upon deployment of new release
Team Members	Meeting, Emails	Changes to be made, known issues and bugs to	After change has been confirmed and approved by the

		be fixed. Deadline of release.	CCB.
Investors and Sponsors	Meetings, Emails	Changes to be made, bugs to be fixed	1 week prior to version release.

6. GLOSSARY

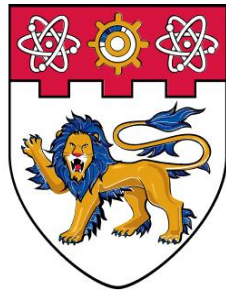
7. ACRONYMS

SRS System Requirement Specification

QA Quality Assurance

CCB Change Control Board

8 APPENDICES



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

PreSchool GoWhere

Test Plan

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Alfred Chow

Andrew Teo

Ryan Tan

Team WannabeFAANG

Version History

Version	Date	Author	Description	Approved by
1.0	19/10/22	Andrew Teo	Initial Draft	Ryan Tan
1.1	27/10/22	Ryan Tan	Working draft 1	Alfred Chow
1.2	28/10/22	Ryan Tan	Final Draft 1	Andrew Teo

Table of Contents

1. Test Plan Identifier	5
2. Introduction	6
3. Test Items (Functions)	6
4. Software Risk Issues	7
5. Features to be Tested	9
6. Features not to be Tested	10
7. Approach (Strategy)	10
7.1.1 White-box unit/ end-to-end testing	10
7.1.2 Regression testing	10
7.2 Prioritisation of testing	11
7.3 Processing untestable elements in requirements and design	11
8. Item Pass/Fail Criteria	11
9. Suspension Criteria and Resumption Requirements	13
9.1 Suspension Criteria	13
9.2 Resumption Requirements	13
10. Test Deliverables	13
11. Remaining Test Tasks	14
12. Environmental Needs	14
12.1 System and Hardware	14
12.2 Documentation	14
12.3 Personnel	14
12.4 Setup	15
13. Staffing and Training needs	15
14. Responsibilities	15
15. Schedule	17
16. Planning Risks and Contingencies	18
16.1 Sudden addition or removal of use cases during testing phase	18
16.2 Insufficient Time to complete testing	18

16.3 Insufficient training to use resources required	19
17. Approvals	19
18. Glossary	20
19. References	20

1. Test Plan Identifier

The test plan identifier serves as a unique identifier for test related documents.

There are 3 different types of test plan:

1. Level Specific test plans: Unit, integration, system and acceptance test plans
2. Type specific test plan - functional, performance, usability and automation test plans
3. Master test plan: comprehensive QA Test plan which contains high level information and a detailed structure of the test process. This test plan rarely undergoes any form of change or revision

Test Plan Name	PreSchoolGoWhere Testing TP1
Test Plan Type	Master Test Plan
Test Plan Version	1.0
Year of Creation	2022

2. Introduction

The procedures and approaches for testing our full stack web application, PreSchool GoWhere , are described in this report. We hope to fulfill the following objectives with this test plan:

1. Identification of the software's functional features that will be tested.
2. Detailed description of our approach and testing methodologies that are being used.
3. Standards used to access whether a test case passes or fails
4. Identify suspension criteria and resumption requirements
5. Identify various types of resources needed for this project
6. Identify various risks and contingencies associated with the project

Unit testing, integration testing, and system testing will be the 3 levels at which we will be applying testing approaches. As part of the unit testing phase of software development, small testable components of an application, known as units, are separately and independently examined to ensure that they perform their intended functions individually. For integration testing, components of the software application are combined and tested as a whole. The goal of this testing level is to identify interface flaws between the modules and functions. Lastly, the system testing tests the complete application as a whole to ensure that the software meets the outlined requirements and upholds the quality standard set.

We will be applying testing techniques to the 3 levels as mentioned above. Blackbox testing and whitebox testing will be used as our primary testing methods.

3. Test Items (Functions)

The following list of items are to be tested:

1. Website Application, PreSchool GoWhere.
2. Quality Plan
3. Software Requirement Specification to ensure traceability

4. Software Risk Issues

These are the list of software risk issues associated with our project::

1. **Availability of external services/libraries used by our software app.**

Our project relies heavily on react google maps service. There is a very small possibility that the google maps service becomes unavailable or requires some significant updates for one of our application features to function as expected.

2. **Government regulation and rules.**

Since PreSchool GoWhere makes use of all the preschools information in Singapore to offer a platform for parents to find a suitable preschool, there may be some rules in the future that limits the use of certain information on Singapore Preschools.

3. **Budget Changes.**

Depending on how the traffic inflow of our application changes, we may need to scale up the website to meet the demands and this may affect our budget estimates. It is important to maintain the reliability and availability of the software service to all users.

4. **Impacts on Client**

Clients create accounts to use PreSchool GoWhere's services. There is a possibility of a security exploit in PreSchool GoWhere which may lead to a loss of accounts, as well as any forms of information given by the client when they create an account on the web application. Therefore it is highly crucial to conduct extensive security testing to ensure that the web application as robust and battle-tested as possible.

5. Features to be Tested

The following features can be further divided into individual test cases and are either unit level testing features or integration level testing features.

Function	Description
Sign up	The user must be able to create an account using their email
Login	The user with an existing account must be able to login successfully. The system must authenticate the user based on his email and password details
Forgot Password	The user must be able to reset the password via a OTP, should he forgets his password
View Available Preschools	The user must be able to view all available preschools displayed on a visual map.
Filter Preschools	The user must be able to use a filter bar to input filter conditions. The system will be able to take in the user's filter conditions and filter out preschools that match the user's filters.
Email Report	The user must be able to request for a full detailed email of preschool(s) that they are interested in.
Saved Filters	The user will be able to save the filter that they have used and access the saved filter whenever they need to.

6. Features not to be Tested

All features mentioned in the Use Case Model of project PreSchool GoWhere are to be tested.

7. Approach (Strategy)

7.1.1 White-box unit/ end-to-end testing

White box testing will be used to verify input-output flow. This will allow QA testers to have a better understanding of how the data flows from the front end into the backend database, alongside the returning of the results to the user on the front end.

Control flow testing will also be conducted to evaluate the the navigation of the web applications. QA Engineer will design predefined paths which will be implemented as test cases for control flow testing. The main aim is the discover if there will be any errors should these test cases be implemented, as well as possible improvements towards the control flow of the application.

7.1.2 Regression testing

Due to the incremental nature of the Agile Software Development lifecycle that has been adopted by the project group, Regression testing fits well into the software development lifecycle.

QA team will be responsible for performing both unit as well as end-to-end testing whenever there has been a new update or bug fix towards the codebase.

7.2 Prioritisation of testing

The QA team will first prioritise testing of components involved in previous bug reports to verify the fixture before proceeding to test other features which were not present in the bug reports filed.

Within all the bug fixes to be tested, the QA team will further prioritise the fixture based off the severity of the issue. Issues with higher severity and can potentially compromise the system will be tested first over issues with lower severity. Issues with higher severity will also be more rigorously tested over those that pose little or no danger to the system.

7.3 Processing untestable elements in requirements and design

Elements in the requirements and design that do not make sense or are untestable with first be pushed to the QA team backlog. Once all testing and report generation has been completed, the QA manager will notified of the untestable elements, who will then update the project manager. Project manager will be in charge of handling the issue for untestable elements by discussing with any relevant parties e.g. development team.

8. Item Pass/Fail Criteria

Features	Pass Criteria	Fail Criteria
Sign up	The user can successfully create an account with an email, provided that there is no other account created with the email they are using	The user is unable to create an account despite using an email unique to the application
Login	If the user has already created an account, they should be able to login to the web application.	The user is unable to login to the web application despite having an account created.

Forgot Password	The user should be able to reset password for their account.	The user is unable to reset their account's password
View Available Preschools	The user should be displayed with a list of available pre schools marked on the map interfaces. Hovering on each map marker will reveal relevant information pertaining to the preschool.	Available pre schools do not display on the map as they should.
Filter Preschools	<p>The user should be a displayed with a filter bar upon clicking on the filter bar button.</p> <p>User should be able to select filters for preschools in accordance to their preference.</p> <p>The web application should display the results that fit within the filter criteria</p>	<p>The user is unable to view the filter bar upon clicking the filter bar button</p> <p>The user is unable to perform selection on any of the filters provided on the filter bar.</p> <p>The web application is unable to load accurate results that fit within the range of the filter criterias.</p>
Email Report	<p>Upon clicking the 'Request Report' button, the user should be able to receive a list of pre-schools alongside relevant information about them.</p> <p>Pre-schools information should fit within the criteria that the user has specified.</p>	<p>The user does not receive any email after clicking the 'Request Report' button.</p> <p>Report sent to the user does not contain the information that has been requested by the user.</p>
Saved Filters	User should be able to save the filters that they have used, and reuse the filters whenever needed.	User is unable to save and reuse filter.

9. Suspension Criteria and Resumption Requirements

9.1 Suspension Criteria

If the test case fails, test for remaining features in PreSchool GoWhere will be terminated. Logging of the failed test case will be done, including relevant informations such as description of error, steps to reproduce error, expected and actual output. The failed test case will then be handled over to the development team for inspection. This is to prevent wastage of resources and possibility of defects being caused by ghost errors that may be caused by earlier defects that were ignored.

9.2 Resumption Requirements

Once the root cause of the test case failure has been identified and resolved by the development team, the testing of remaining features will be continued. Features that are independent of the features include in the failed test cases will continue to be tested regardless of the test case failure. This will ensure that there will not be any unnecessary backlogging of testing.

10. Test Deliverables

Deliverables	By	Deadline
Test Plan Document	Ryan Tan	15th Oct 2022
Test Cases	Andrew Teo	16th Oct 2022
Test design specification	Andrew Teo	16th Oct 2022
Tools and their output	Ryan Tan	18th Oct 2022
Simulators	Ryan Tan Andrew Teo	18th Oct 2022

Static and dynamic generators	Andrew Teo	20th Oct 2022
Error logs and execution logs	Ryan Tan	20th Oct 2022
Problem reports and corrective action	Ryan Tan Andrew Teo	21st Oct 2022

11. Remaining Test Tasks

12. Environmental Needs

12.1 System and Hardware

- Personal computer with Javascript-compatible Internet browser
- Supabase Server
- Access to internet connection

12.2 Documentation

- Reference Documents (Use Case Model, Quality Plan)
- Test Cases
- Bug Reporting tools

12.3 Personnel

- Developers
- QA manager

- QA engineer

12.4 Setup

To test for the ability for our web application to work on various devices as well as its ability to be mobile responsive, it will be tested on both PC as well as mobile phones of different models

13. Staffing and Training needs

The following staffing will be required for the test plan

Role	Quantity
QA manager	1
QA engineer	1
Lead developer	1

The following softwares will be required to be included in training

- Jira
- Trello
- Selenium Webdriver
- Jest

14. Responsibilities

Role	Personnel Name	Tasks
QA Manager	Andrew Teo	<ul style="list-style-type: none">- Oversees the entire testing process- Decides on priority

		<p>of functionalities to test on</p> <ul style="list-style-type: none"> - Decides on technology used for testing
QA Engineer	Ryan Tan	<ul style="list-style-type: none"> - Design testing procedures - Generate test plan - Evaluation of test plan - Writing of automated test scripts
Database Testing	Dhanyamraju Harsh Rao Malavade Sanskar Deepak	<ul style="list-style-type: none"> - Execution of backend testing of database - Storing and logging of database test results
Frontend Testing	Alfred Chow Parashar Kshitij	<ul style="list-style-type: none"> - Execute unit and end to end test on functionality of the website
Database Administrator	Dhanyamraju Harsh Rao Malavade Sanskar Deepak	<ul style="list-style-type: none"> - Ensures database uptime during testing - Management of test data
Deployment Testing	Alfred Chow Parashar Kshitij	<ul style="list-style-type: none"> - Ensures that application is able to be deployed - Testing of deployed

		applications, checks if the application is functioning as per expected when hosted on cloud.
--	--	---

15. Schedule

Task	Estimated time taken	Deadline
Design tests	1 day	15th Oct 2022
Implement test	3 days	16th Oct 2022
Execute Tests	1 days	16th Oct 2022
Generating test report for developers	2 days	18th Oct 2022
Evaluate tests/ Fix bugs	2 days	18th Oct 2022
Re-run tests after resolution of errors	1 day	20th Oct 2022
Documentation	2 day	20th Oct 2022

16. Planning Risks and Contingencies

16.1 Sudden addition or removal of use cases during testing phase

Risk:

During the testing phase, there is a sudden additional or removal of originally specified use case.

Contingency:

Due to the uncertainty in whether the use case will be added or removed. QA team will backlog testing for the particular functionality and proceed to test on the other remaining functionalities.

Should the need arise, the QA team will also have to redevelop and redesign test cases if there are other functionalities dependent on the removed or added functionality.

16.2 Insufficient Time to complete testing

Risk:

Team is unable to meet the deadline for deliverables, leaving QA team with insufficient time to conduct intensive testing.

Contingency:

Usage of the Agile Software Development model is an effective combat to the possibility of insufficient testing time from showing up as features

will be tested in increments.

However, under the worst case scenario:

QA manager will be in charged of deciding which test cases and functionalities to prioritise for testing over the others, and will then update the Project Manager.

Test cases will also be reduced.

16.3 Insufficient training to use resources required

Risk:

Personnels involved in software testing might not be sufficiently experienced to handle to software tools required.

Contingency:

Evaluation tests should be done for personnels required to use software tools for testing. Only allow them to perform testing once they are adequately proficient in software usage. More training can be done too.

17. Approvals

The project manager will be in charge of approving the tests designed by both the QA manager as well as the QA Engineer. Once the approval has been passed, the QA engineer will proceed to implement and execute the tests.

QA Engineer will be in charge of generating any bug fixing reports and reporting to the QA manager subsequently.

QA Manager will then update the Project Manager on the status of the tests.

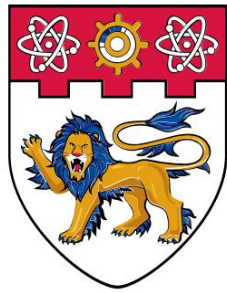
18. Glossary

OTP: One Time Password

QA: Quality Assurance

19. References

- Project Plan
- Use Case Model
- Quality Plan
- System Requirement Specification
- Risk Management Plan
- Test Cases and Requirements Coverage



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

PreSchool GoWhere

TEST CASES AND REQUIREMENTS TEST COVERAGE REPORT

Parashar Kshitij

Dhanyamraju Harsh Rao

Malavade Sanskar Deepak

Alfred Chow

Andrew Teo

Ryan Tan

Team WannabeFAANG

Version History

Version	Date	Author	Description	Approved by
1.0	19/10/22	Andrew Teo	Initial Draft	Ryan Tan
1.1	24/10/22	Ryan Tan	Working draft 1	Alfred Chow
1.3	26/10/22	Alfred Chow	Working Draft 2	Andrew Teo

Test Case 1 - Login Authentication

Test case ID: tc1	Test Designed by: Andrew
Test priority: High	Test Design Date: 15/10/22
Title: Login Authentication	Test Executed By: Andrew
Description: Users with an existing account can login successfully into our web application.	Tested on: 21/10/22
Pre-conditions: User must have an existing account with our web application	
Dependencies: N/A	

Steps:

Test Step	Test Input	Expected Result	Actual Result	Status
(1) Enter the url of the web application in a browser	N/A	User should see the login page	User is redirected to the login page	Success
(2) Enter valid email in the email field	mapogan219@corylan.com	User should see his email in the input field	Email field in the login page displays 'mapogan219@corylan.com'	Success
(3) Enter valid password in the password field	hellopassword1	User should see his password being hidden in the field	Password field shows "●●●●●●●●●●", which hides the text of the pw.	Success
(4) Click "Sign In" button	N/A	User should see a "successfully logged in" popup message and get redirected to the home page.	"successfully logged in" popup message appears and user is being redirected to the home page	Success

** The user must enter the correct matching email and password. Otherwise, the web application will display a "login failed" alert message and the user is then prompted to re-enter his email and password details.

Test Case 2 - Sign up new account

Test case ID: tc2	Test Designed by: Ryan
Test priority: High	Test Design Date: 15/10/22
Title: Sign up a new account	Test Executed By: Ryan
Description: Users without a pre-existing account can sign up for a new account with our web application.	Tested on: 21/10/22
Pre-conditions: User must use an email that has not been registered with our web application.	
Dependencies: N/A	

Steps:

Test Step	Test Input	Expected Result	Actual Result	Status
(1) Enter the url of the web application in a browser	N/A	User should see the login page	User is redirected to the login page	Success
(2) User clicks the "Sign up" link below the login form	N/A	User should see the signup page	User is redirected to the signup page	Success
(3) Enter valid email in the email field	helloJack1@gmail.com	User should see his email in the input field	Email field in the login page displays 'helloJack1@gail.com'	Success
(5) Enter valid password in the password field	Password@12	User should see his password being hidden in the field	Password field shows "●●●●●●●●●●", which hides the text of the pw.	Success
(6) Enter the same password in the confirm password field	Password@12	User should see his password being hidden in the field	Password field shows "●●●●●●●●●●", which hides the text of the pw.	Success

(7) Click on "generate OTP"	N/A	User should see a "OTP successfully sent to email" popup message.	User should see a "OTP successfully sent to email" popup message.	Success
(8) Enter OTP (sent to email)	N/A	User should see his OTP in the field	OTP field shows the otp entered by the user	Success
(9) Click on "Sign up" button	N/A	User should see a "successfully signed up" popup message and get redirected to the home page.	"successfully signed up" popup message appears and user is being redirected to the home page	Success

Alternate Flows:

Actions	Test Input	Expected Result	Actual Result	Status
User enters a mismatch of passwords in the password and confirm - password fields	Password : "password@1" Confirm Password: "password@3"	User should see an alert message, which tells the user that passwords do not match.	Alert message shows "Sign up unsuccessful", "Passwords do not match. Please try again!"	Success
User enters invalid email Input	Email: "hellothisismyemail"	User is prompted to enter a valid email	A pop up message shows up, telling user that the input is invalid	Success
User enters invalid otp (different from the otp sent to email)	N/A	Alert message shows that sign up is unsuccessful and user is prompted to register his details again	Alert message shows that sign up is unsuccessful and user is prompted to register his details again	Success

Test Case 3 - Filter Preschools

Test case ID: tc3	Test Designed by: Alfred
Test priority: High	Test Design Date: 15/10/22
Title: Filter preschools	Test Executed By: Alfred
Description: User can use a filterbar which provides a set of search filters for the user to find a preschool that matches his search criterias.	Tested on: 21/10/22
Pre-conditions: User must allow location services for the distance filter to work.	
Dependencies: N/A	

Steps:

Test Step	Test Input	Expected Result	Actual Result	Status
(1) User expands the filterbar	N/A	Filterbar consisting of multiple search filters appears	Filterbar consisting of multiple search filters appears	Success
(2) User enters some of the filter criterias	Distance: "20km" Second Language offered: "Chinese"	Filter options displays the updated values selected by the user	Filter options displays the updated values selected by the user	Success
(3) User clicks the "Filter" button	N/A	"Loading filters" message appears and google maps displays the preschools that match the search criterias. (Each preschool is represented by a marker on the map)	"Loading filters" message appears and google maps displays the preschools that match the search criterias.	Success

(4) User clicks on a preschool	N/A	A popup message above the preschool will appear, consisting of the basic information of the preschool	A popup message above the preschool marker appears, displaying the basic information of the preschool	Success
--------------------------------	-----	---	---	---------

Alternate Flows:

Actions	Test Input	Expected Result	Actual Result	Status
User disabled his location services but uses the distance criteria as one of the filters	N/A	A alert message will display, prompting the user to enable his location services	A alert message displays, prompting the user to enable his location services	Success

Test Case 4 - Save Filters

Test case ID: tc4	Test Designed by: Andrew
Test priority: Medium	Test Design Date: 15/10/22
Title: Save Filters	Test Executed By: Andrew
Description: Users can save his current filter input options and load them another time.	Tested on: 21/10/22
Pre-conditions: User is in the home page which shows the google maps display	
Dependencies: N/A	

Steps:

Test Step	Test Input	Expected Result	Actual Result	Status
(1) User expands the filterbar	N/A	Filterbar consisting of multiple search filters appears	Filterbar consisting of multiple search filters appears	Success
(2) User enters inputs for the filter criterias	<u>Distance:</u> "20km" <u>Second Language offered:</u> "Chinese"	Filter options displays the updated values selected by the user	Filter options displays the updated values selected by the user	Success
(3) User clicks the "Save Filter" button	N/A	Popup message will appear, displaying "Successfully saved filters!"	Popup message appears, displaying "Successfully saved filters!"	Success
(4) User clicks the "Use Saved Filters" button	N/A	Filterbar gets updated with the filter options that the user saved previously	Filterbar gets updated with the filter options that the user saved previously	Success

Alternate Flows:

Actions	Test Input	Expected Result	Actual Result	Status
User has not saved any filters before but clicks "Use Saved Filters" button	N/A	Warning message appears, warning the user that "You have not saved any filters previously!"	Warning message appears, warning the user that "You have not saved any filters previously!"	Success

Test Case 5 - Send Email Report

Test case ID: tc5	Test Designed by: Ryan
Test priority: High	Test Design Date: 15/10/22
Title: Send Email Report	Test Executed By: Ryan
Description: Sends a detailed email report on the preschools upon the user's request	Tested on: 21/10/22
Pre-conditions: User has filtered a set of preschools that matches his search criterias, and now wants a email report on these preschools	
Dependencies: N/A	

Steps:

Test Step	Test Input	Expected Result	Actual Result	Status
(1) User clicks the "Request Report" button in the filter bar	N/A	Popup message appears, saying "Email report successfully sent!"	Popup message appears, saying "Email report successfully sent!"	Success
(2) User checks the inbox of the email used to register with our web application	N/A	A report on the preschools is sent to the user's email within 3s. Email report consists of information such as the preschools' center code, name, contact number, email address and website	User receives an email report on the preschools within 3s. Email report consists of information such as the preschools' center code, name, contact number, email address and website	Success