NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

School of Computer Science and Engineering

Advanced Software Engineering

Project Plan

Website Portal: HouseHunt

Supervisor: Althea Liang

13 September 2021

By Team 5:

Tang Kai Wen, Alvin

Bankata Mishra Spriha

Heng Chor Chen, Sabrina

Yong Wen Shiuan

Gupta Suhana

Heng Fuwei Esmond

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	Sep 23th, 2021	TANG KAI WEN, ALVIN	First Version
1.1	Sep 28th, 2021	GUPTA SUHANA HENG FUWEI ESMOND TANG KAI WEN, ALVIN	First Version Updates
2.0	Oct 1st, 2021	YONG WEN SHIUAN TANG KAI WEN, ALVIN	Refactored Unadjusted Function Point calculation
2.1	Oct 4th, 2021	HENG FUWEI ESMOND	Minor Version Updates
3.0	Oct 5th, 2021	GUPTA SUHANA TANG KAI WEN, ALVIN	Formatting Report for Submission

Table of Contents

1. Introduction

1.1. Project Overview

The HouseHunt project is an intelligent agent providing assistance to users via providing the Ideal Resale HDB system. This web based flat resale tool estimates the value of real estate sellers' flats based on factors such as flat type and model, locality, remaining lease, and floor area. HouseHunt also caters to potential buyers searching for their own HDB accommodation.

1.2. Project Description and Scope

As a web-based application, HouseHunt seeks to be accessible to both buyers and sellers without any specialised equipment, only requiring an internet connection and a smart device or computer.

Singapore's real estate prices are highly dependent on a multitude of factors, which can be overwhelming to novice users. Moreover, the recent pandemic has caused a wave of cancellation of Build-to-Order. As a result, even experienced users may face confusion with regards to the impact it has on the resale market. Therefore, there is a need for a comprehensive and cogent solution that collates currently available real estate information and presents it based on users' priorities. HouseHunt aims to be the one-stop solution to the average Singaporean's real estate needs by focusing on ease of access and reliability.

HouseHunt assists users by simplifying the vast quantity of information available online or given to them by real estate agents. The information it provides includes flat finding based on certain criteria, estimations of resale flats prices based on recent data, house affordability calculations using user's income information, and data visualisations of house prices against certain parameters such that users can understand the pricing trends in Singapore. This will help in avoiding decision fatigue and help buyers in purchasing a HDB flat that they are satisfied with. 'HouseHunt,' also estimates the value of real estate sellers' flats based on factors such as flat type and model, locality, remaining lease, and floor area.

Overall HouseHunt helps connect buyers and sellers and simplifies the complex Singaporean real estate market for the benefit of both parties.

2. Project Organization

2.1. Team Structure

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

- Project Manager: Tang Kai Wen, Alvin
- Lead Developer: Yong Wen Shiuan
- Quality Assurance Manager and Engineer: Bankata Mishra Spriha
- Front End Developer: Gupta Suhana
- Back End Developer: Heng Fuwei, Esmond
- Release Manager and Engineer: Heng Chor Chen, Sabrina

2.2. Roles and Responsibilities

Project Manager: Tang Kai Wen, Alvin

- Oversees project progress
- Approves and executes project plan
- Assigns tasks and reports status of project to team members
- Manages and motivates team members
- Represents the team to the outside world

Lead Developer: Yong Wen Shiuan

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

Quality Assurance Manager and Engineer: Bankata Mishra Spriha

- In charge of overall product and process quality
- Responsible for preparation of quality documentation and reports
- Ensures software is of acceptable quality
- Comes up with testing strategies
- Verifies software requirements align with user requirements
- Carries out test procedures

- Develops and manages test plans

Front End Developer: Gupta Suhana

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

Back End Developer: Heng Fuwei, Esmond

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

Release Manager and Engineer: Heng Chor Chen, Sabrina

- Creates baseline and conducts release reviews
- Ensures application releases are delivered on time and within budget by measuring and monitoring progress
- Builds and integrates changes for delivery

2.3. Team Communication

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

- Weekly meetings are held on Tuesdays.
- Google Drive will be used for file sharing between members
- Project updates and other meetings are announced in the Telegram group chat
- The team's Wiki page will be used for the uploading of finalized documents
- Split up into subgroups as necessary, in order to work more cooperatively on specific problems (eg. documentation and developers).

3. Process Definition

3.1. Lifecycle Model

Throughout this HouseHunt project we intend to use the Agile Model as this methodology is more flexible than the traditional Waterfall Software Development Life Cycle (SDLC) due to the iterative approach towards software development. With the short timeline available for the development of HouseHunt, using an Agile Model will help to ensure that the end product is of high quality. The Agile method can achieve all these by involving design, coding, unit testing, integration and quality assurance into the SDLC.

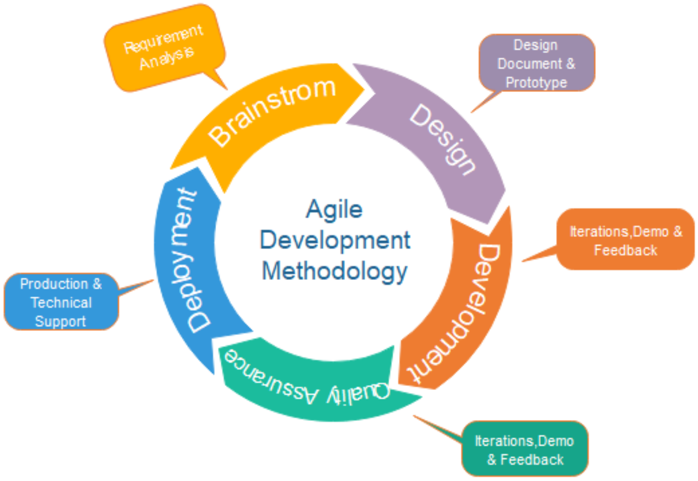

Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Agile Development Methodology
Iterations,Demo & Feedback
Development
Production & Technical Support
Deployment
Quality Assurance
Iterations,Demo & Feedback

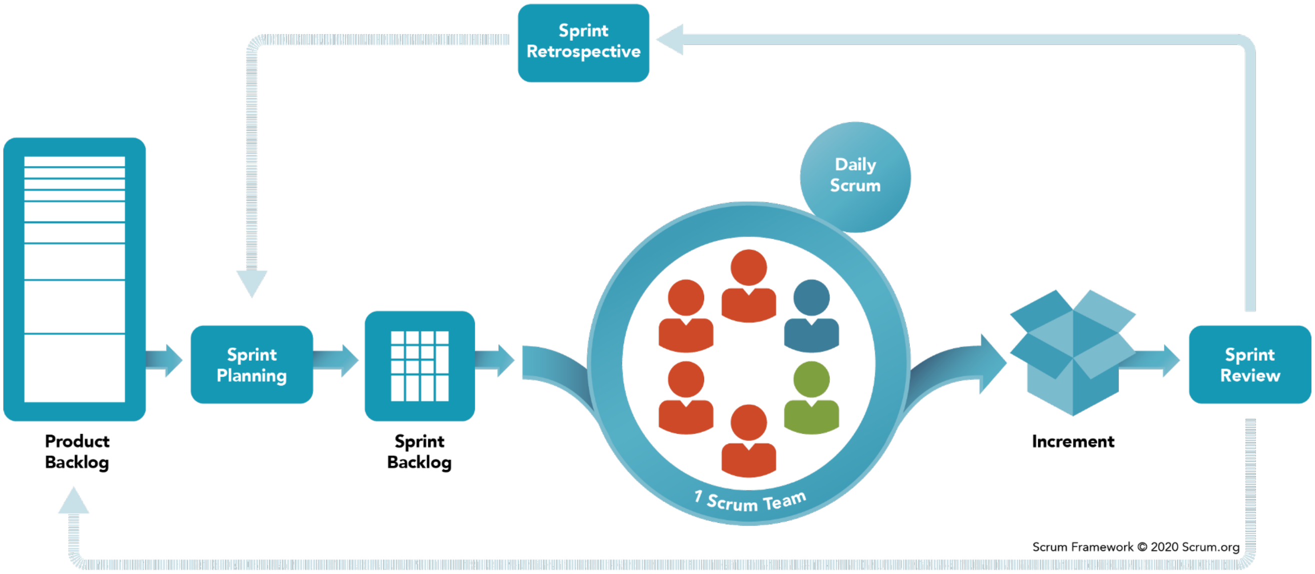
Fig. Agile Model

The first iteration of HouseHunt will be released as indicated on the Estimations section of this document. Agile Model focuses on client engagement, the client will be asked to evaluate each iteration of the system so that improvements can be made on the next iteration.

Before the release of each iteration, thorough testing will be carried out. Ensuring the product is of high quality, at the same time it also allows for faster development.

For this project we will be using the SCRUM framework; SCRUM is an iterative and incremental framework for managing product development featuring a SCRUM team which doubles in our case as our main development team as well. The three core roles in the SCRUM team are:

Product Owner	Create the product backlog (To-Do-List) and set the task priorities to deliver the highest value
Scrum Master	Facilitates Scrum process and prevent distractions
Development Team	Self-Organizing group that takes on and determines how to deliver set tasks



Our team seeks to work with SCRUM mainly because of the 'Sprints' and the daily meetings we have online on the project. This is due to other commitments that our team has during this busy period of time. A sprint is when a usable and potentially releasable product increment is created. These sprints usually last 1-2 weeks for the HouseHunt projects as our deliverables are set to be by that time frame. The workflow for each sprint is as follows:

Sprint Planning	The development team selects tasks to do from the backlog and plan how to achieve the completion of the tasks. A Sprint Goal for them will be set for the team to target and accomplish it
Daily Scrum	Development Team will meet every day during the Sprint to check the progress towards the Sprint Goal
Sprint Review	The team completes testing on the tasks and discuss what was done, and make changes accordingly to the backlog
Sprint Retrospective	Development Team will analyze what went right, what could fair better, and how it can be further improved

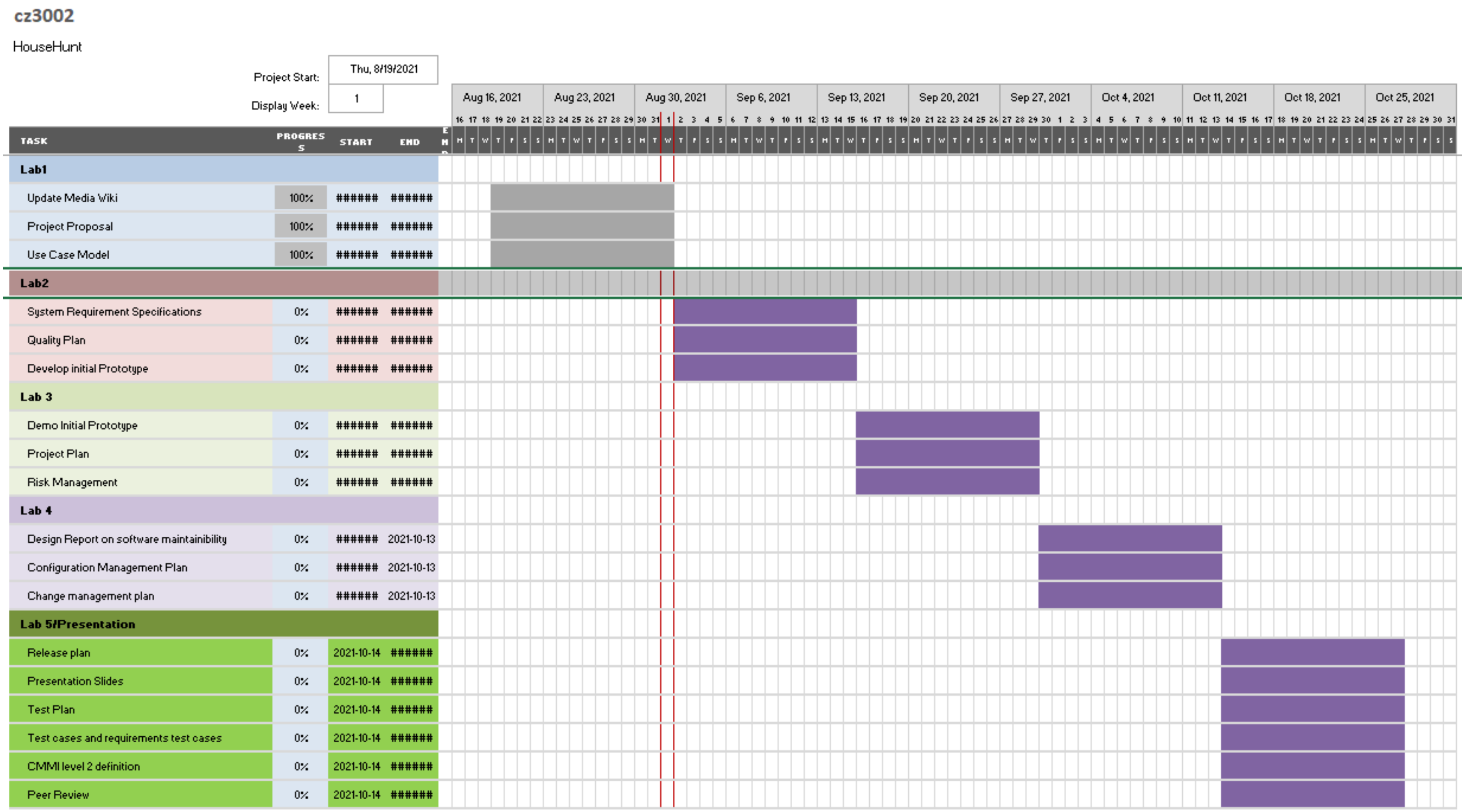
Once a Sprint is done, the whole process repeats with the Development Team picking another product backlog to work on.

Other reasons why we selected this model is because:

- Subdivision of the project into easily manageable parts during the sprints and tasks can be assigned to project members clearly and effectively.
- Testing is done immediately when the tasks are completed for the sprint reviews.
- It allows the project to be fast paced to ensure time is managed efficiently to meet the tight schedule given to HouseHunt
- Constant feedback from stakeholders is obtained to change the project according to their demands.
- There is transparency in the amount of effort each member puts into the project.

4. Schedule

4.1. Activity Dependencies and Schedule



cz3002

HouseHunt

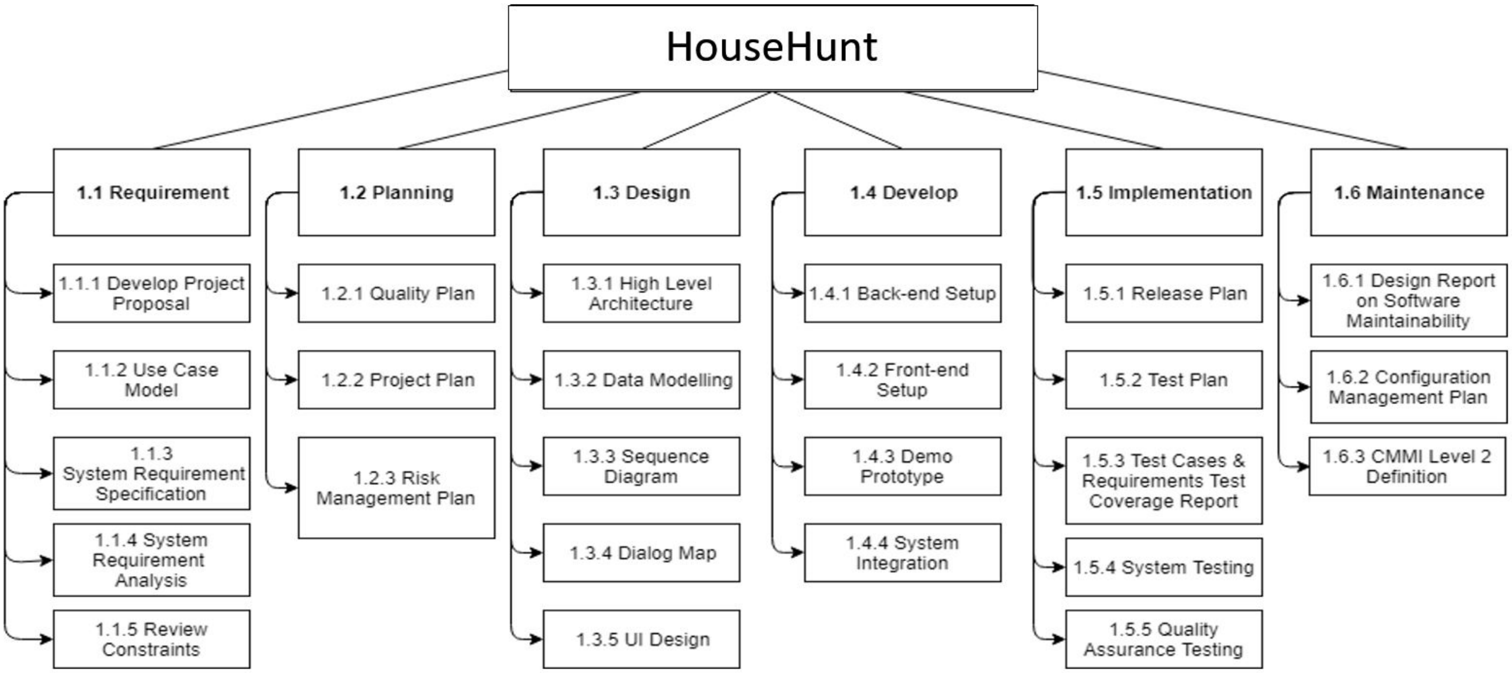
Project Start: Thu, 8/19/2021

Display Week: 1

TASK	PROGRESS	START	END
Lab1			
Update Media Wiki	100%	######	######
Project Proposal	100%	######	######
Use Case Model	100%	######	######
Lab2			
System Requirement Specifications	0%	######	######
Quality Plan	0%	######	######
Develop initial Prototype	0%	######	######
Lab 3			
Demo Initial Prototype	0%	######	######
Project Plan	0%	######	######
Risk Management	0%	######	######
Lab 4			
Design Report on software maintainibility	0%	######	2021-10-13
Configuration Management Plan	0%	######	2021-10-13
Change management plan	0%	######	2021-10-13
Lab 5/Presentation			
Release plan	0%	2021-10-14	######
Presentation Slides	0%	2021-10-14	######
Test Plan	0%	2021-10-14	######
Test cases and requirements test cases	0%	2021-10-14	######
CMMI level 2 definition	0%	2021-10-14	######
Peer Review	0%	2021-10-14	######

Figure 1: Project Schedule

4.2. Work Breakdown Structure



4.3. Work Packages

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

1. Requirement Specification
2. Project Plan
3. User Interface Designing
4. Application Development
5. Coding & Unit Testing
6. Maintenance & Quality Assurance

4.4. Activity Dependencies

The following table describes the dependencies of the deliverable work packages based off the critical path of HouseHunt Project:

Work Package #	Work Package Description	Duration (Days)	Dependencies
1	Requirements	15	--
2	Project Plan	9	--
3	User Interface Designing	6	--
4	Application Development	10	1,2,3
5	Coding & Unit Testing	8	4
6	Maintenance & Quality Assurance	13	5

4.4.1. Activities

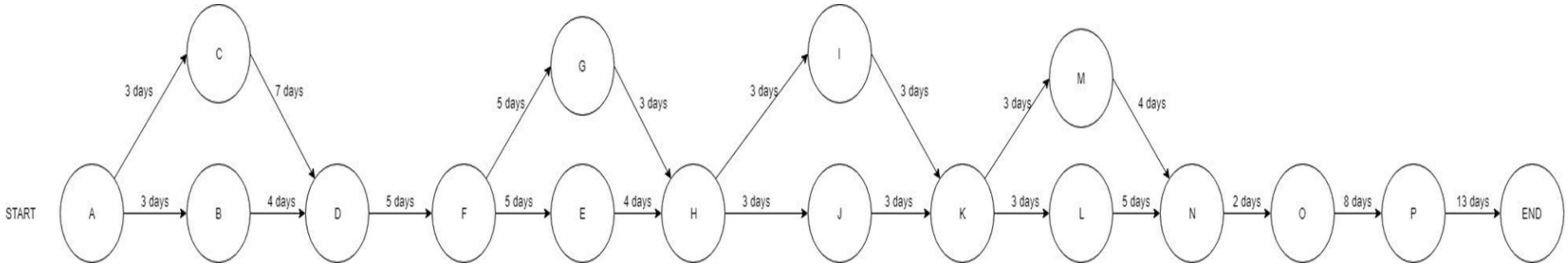
After further review on finalizing the project schedule based on the past few deliverables submission. Our team has updated the duration for the activities accordingly to provide a clearer overview of how our team will carry out the project plan of HouseHunt

Table for Activities		
	Activities	**Duration**
Requirements	A. Identify end-user	3 days
	B. Interviewing customers	4 days
	C. Conduct survey	7 days
	D. Analyze survey results and gather feedbacks	5 days
Project Plan	E. Quality Planning	4 days
	F. Project Planning	5 days
	G. Risk Management Planning	3 days
	H. Design UI Mockup	3 days
	I. Design System Architecture	3 days

User Interface Designing	J. Source for software technologies to be used to develop the project	3 days
Application Development	K. Build database	3 days
	L. Build front-end	5 days
	M. Build back-end	4 days
	N. Integrate database, front-end and back-end	2 days
Coding & Unit Testing	O. Validate the entire system through system testing	8 days
Maintenance & Quality Assurance	P. System Maintenance till handover (Software upgrades, bug fixes, Software documentation for maintainability)	13 days

4.4.2. Activity Network Diagram

The following Activity Network Diagram describes the above in more graphic detail. For a clearer view of the Activity Network Diagram, refer to the uploaded file - HouseHunt_Activity_Network_Diagram.jpg



4.4.3. Critical Path

A → C → D → F → E → H → I → J → K → L → N → O → P → END (61 days)

4.4.4. Time Cost Model

We came up with the time cost model shown below, after considering all aspects of our system.

Activities	Crash cost per week	Crash days available
D → F	$1000	3
C → D	$400	2
L → N	$1200	1
O → P	$300	2

The reasons for the above crash cost for each activities are as justified below after vigorous brainstorming and analysis done to ensure the final product - HouseHunt - retains the same quality as it would if we kept to the original schedule

Activities	Activity’s Name	Reasons for Crashing
C → D	Conduct Survey	Our team is able to use money as an incentive to obtain quality feedbacks faster
D → F	Analyzing survey results and gather feedback	With higher budget available, we will be able to use analytic models to better identify the key takings faster
L → N	Build Front-End	Hiring a better team will shorten the duration for when the front-end will finish
O → P	Validate the entire system through system testing	An increase in budget allows better software to analyze the system faster

After having a meeting with the stakeholders of the project, they have decided to budget $500 per day saved, up to a maximum of $3000, in order for the project to be able to be completed earlier. Based on the Time-Cost Model, we have worked out the most efficient method to shorten the project with the given budget.

1. Reduce 1 day from O → P (Total cost: $300, Profits: $200, (O → P remaining: 1))
2. Reduce 1 day from O → P (Total cost: $600, Profits: $400, (O → P remaining: 0))
3. Reduce 1 day from C → D (Total cost: $1000, Profits: $500, (C → D remaining: 1))
4. Reduce 1 day from C → D (Total cost: $1400, Profits: $600, (C → D remaining: 0))

In order to efficiently maximise the profits for our team while reducing the project duration, our team has finalized it is best to crash the project by 4 days to allow the maximum profits earned of $600. This brings down the overall project duration to 57 days

4.5. Work Package Details

Work packages are listed below. A team member, indicated in bold, has been assigned as primarily responsible for each work package and will coordinate that package. Most of the team members are involved for all the portions.

Project	HouseHunt
Work Package	1 – Requirements (1/6)
Led By	**TANG KAI WEN, ALVIN**
Effort	13 Days
Start Date	29 August 2021
Purpose	This package involves gathering of customer requirements, as well as defining the problem statement and brief objectives to achieve the desired functionality of the project.
Inputs	Customer's Requirements
Activities	Identifying the end-users, interviewing the customers, doing surveys and gathering feedback on the problem defined.
Outputs	A written document of the Requirements Specification.

Project	HouseHunt
Work Package	2 – Project Plan (2/6)
Led By	**BANKATA MISHRA SPRIHA**

Effort	9 Days
Start Date	05/01/21
Purpose	This package helps the team to plan for the project in 3 pain aspects the main Project Plan, Quality planning and assurance as well as identifying potential risk and planning for them
Inputs	None
Activities	Gathering information and detailing it in the documents: Quality Planning, Project Plan as well as Risk Management Plan. Determining the Quality Model, Software Metrics, Verification and Validation methods to be employed in the development of HouseHunt.
Outputs	A written document of the Project Plan, Quality Plan as well as Risk Management Plan.

Project	HouseHunt
Work Package	3 - User Interface Designing (3/6)
Led By	**GUPTA SUHANA**
Effort	6 Days
Start Date	24/08/21
Purpose	This package involves designing the product to create the prototype.
Inputs	None

Activities	Designing Wireframe, System Architecture, as well as understanding and sourcing for the right software technologies to be used to develop the features in our project. In addition, the system architecture and the various components will be identified to provide information on how they are related. The design will be based on the ISO Model 9241 Human System Interaction which touches on Functionality, Usability, Maintainability, Reliability, Efficiency and Portability these have been detailed out in the Quality Plan.
Outputs	High Level Design and Architectural Specifications

Project	HouseHunt
Work Package	4 - Application Development (4/6)
Led By	**YONG WEN SHIUAN**
Effort	10 Days
Start Date	30/09/21
Purpose	Developing the prototype of the product and required dependencies
Inputs	1 – Requirements, 2 – Planning, 3 – Design Packages
Activities	Involves building the databases, the front-end, back-end and integrating all the components together.
Outputs	The Prototype

Project	HouseHunt

Work Package	5 - Coding & Unit Testing (5/6)
Assigned To	**HENG FUWEI ESMOND**
Effort	6 Days
Start Date	15/10/21
Purpose	Ensure smooth delivery and quality of the product to the customer’s hands.
Inputs	4 – Develop
Activities	Validation of the entire system through system testing and integration which is done using release plans, test plans as well as consistent monitoring and controlling.
Outputs	Test Plan, Release Plan, End-Product

Project	HouseHunt
Work Package	6 - Maintenance & Quality Assurance (6/6)
Assigned To	**HENG CHOR CHEN, SABRINA**
Effort	13 Days
Start Date	08/10/21
Purpose	Ensure the end-product is bug/error-free and does not break.
Inputs	None

Activities	Software upgrades, Repairs, Fixes, Documenting Software Maintainability
Outputs	A written document of the Software Maintainability.

5. Project Estimates

5.1. Code Size Estimation using Function Points

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

5.1.1. Unadjusted Function Points

HouseHunt system supports the following proposed Functional Requirements:

User:

- The System must allow User to input resale flat data to estimate it's selling price
- The System must allow User to view resale price based on town, flat type, or year of sale
- The System must allow User to filter out resale flats based on the flat's requirements
- The System must tabulate User's maximum affordable price from the User's financial inputs

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

Rating Inputs:

- Input fields to filter HDB resale flat transactions
- Navigating on the Google Maps API
- Input financial data to calculate affordability of flat
- Filtering the type of visual data of resale flats

Rating Outputs:

- Displaying a filtered list of the results matching the user's search criteria (flat type, remaining lease in years, maximum resale price, town, minimum floor area (in sqm), flat model)
- Displaying estimate selling price of user's resale flat based on the above criteria
- Displaying data visualisation of resale flats by town, flat type, or year of sale
- Displaying maximum affordable price based on user's finances (monthly income, monthly debt, loan interest rate, cash towards down payment)

Rating Inquiries:

- Selecting the resale HDB flats according to the user's criteria

Rating Logical Files:

- Resale HDB Database

Rating Interfaces:

- Menu Page
- Estimate Page
- Visualization Page
- Search Page
- Calculate Page

The summary of the above analysis is as follows:

Element	Details	Complexity
Inputs	Filter HDB resale flat transactions	Low
	Navigating Google Maps API	Low
	Input financial data	Low
	Filtering Visual Data	Low
Logical Files	Resale HDB Database	High
	Display Filtered Search Criteria	High

Outputs	Display Estimate Resale Flat Price	Low
	Display Visualization	High
	Display Affordability Price	Low
Inquiries	Selecting the resale HDB flats based on the user’s criteria	High
Interfaces	Menu Page	Low
	Estimate Page	Medium
	Visualization Page	Medium
	Search Page	High
	Calculate Page	Medium

Based on the table, the calculation of Unadjusted Function Points is as follows:

Characteristic	**Low Complexity**		**Medium Complexity**		**High Complexity**	
# of Inputs	4	× 3	0	× 4	0	× 6
# of Outputs	2	× 4	0	× 5	2	× 7
# of Inquiries	0	× 3	0	× 4	1	× 6
# of Logical Files	0	× 7	0	× 10	1	× 15
# of Interfaces	1	× 5	3	× 7	1	× 10
Unadjusted FP	25		21		45	
Total=L+M+H	91					

5.1.2. Adjusted Function Points

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

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

4 = Significant influence
5 = Strong influence

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

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

		combinations on one or more internal logical files (count as three items).
Total score	37	
Influence Multiplier = Total score × 0.01 + 0.65 = 37 × 0.01 + 0.65 = 1.02		
Adjusted FP = Unadjusted FP × Influence Multiplier = 91 × 1.02 = 92.82		

5.1.3. Lines of Code

According to Capers Jones statistics, each Function Point requires 53.33 lines of code if the application is implemented using python

Therefore, we have: **Lines of Code** = 92.82 FP × 53.33 LOC/FP = **4950 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.

Our team will be adopting the Top-Down estimation for the calculation:

Top-Down Estimation

- Working days include 5 days in a week.
- Effort = Size / Production Rate = (4950 LOC) / (62 LOC/PD)[1] = 79.84PD
    - [1] - LOC/PD based on Industrial Benchmark, 1997: Canada
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (79.84) $^{1/3}$ = 12.92 Person Day (PD)
- Initial schedule = 79.84 PD / 5 days a week = 15.97 Weeks
- Team size = 79.84 PD / 12.92 PD = 6.18 P = 7 Persons
- Compression rate = obtained team size / calculated team size = 6/7 = 0.86
- Desired Schedule = 15.97 x 0.86 = 13.73 weeks
- Total person-hours (PH) = 79.84 PD × 8 hours = 638.72 PH

5.2.1. Distribution of Effort

1990's Industry Data	Work Package	Distribution	Top-Down Estimates (638.72 Hours)
Preliminary Design 18 %	Project Plan	9%	57.48
	Requirements	9%	57.48
Detailed Design 25 %	User Interface Designing	10%	63.87
	Application Development	15%	95.82
Code & Unit Testing 26 %	Coding & Unit testing	14%	89.42
	Online Documentation	12%	76.65
Integration & Test 31 %	Maintenance & Quality Assurance	31%	198.00
Extrapolated total effort			638.72
2% for project management			12.77
3% for contingency			19.16
Total effort			670.65

These duration estimates are based on the assumption that each team member works an equal amount on any given work package.

5.3. Cost Estimates

Item	Supplier	Quantity	Unit Price	Total
Front-End Developer	-	1	$5,000	$5,000
Back-End Developer	-	2	$5,000	$10,000
Quality Assurance Manager	-	1	$4,000	$4,000
Project Manager	-	1	$4,000	$4,000
Release Manager	-	1	$4,000	$4,000
Computers	DELL	6	$1,000	$6,000
Office Rental	NTU	1	$6,000	$6,000
Router	TP-Link	1	$35	$35

Main Power Supply	PUB	1	$500	$500
			Total	$39,535

***Table 1**: Associated Budget Requirements*

6. Product Checklist

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

Project Deliverable	Estimated Deadline
Project Proposal	01/09/21
Use Case Model	30/08/21
System Requirement Analysis	13/09/21
Initial Prototype with Front-Page and Back-End Database	20/09/21
Quality Plan	12/09/21
Finalized Prototype with Front-Page and Back-End Database	27/09/21
Project Plan	05/10/21
Risk Management	04/10/21
Showcase Completed Prototype	07/10/21
Design Report on Software Maintainability	20/10/21

Configuration Management	14/10/21
Change Management	14/10/21
Release Plan	20/10/21
Showcase Completed HouseHunt Application	21/10/21
Test Cases	19/10/21
Test Plan	19/10/21
CMMI Level 2 Definition	20/10/21

7. Best Practice Checklist

Practices Checklist
All Documentations need to be in a clear and concise format being accessible to all of the team members involved.
We will focus on the requirements, and make sure they are free of ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification of the application and must be secure to keep it out of the wrong hands
We have to keep the documentation as simple as possible and hence any readers will easily understand what is documented. Thus, complexity management is one of the major challenges. Strive to: • Minimize communications between different people, exponential communication, additional costs, modules, procedures and data being needed to be included in the process of creation of the application. • Keep it simple for all the product features and try to be as user- friendly as it is the key importance for creation and, as long as our design meets the customer requirements functionality, it will hence be an easy application for anyone to adapt to without any question or doubt while using the application we created.
We must always visualize what we build and follow according to our project schedule in order to measure our progress and take management action if necessary if any of us were unable to meet the goal we set for the project. For example: A manager must have good communication with his or her colleagues to make sure all the reports,presentation, code, design available for review are in proper order and appropriate towards the needs of the customer before the deadline is met.
Plan for continuous change. We must: • All documented designs, test cases, source code should have revision numbers documented and dates revision history being commented to indicate what are the changes being made so that other developers would understand easily, if there is a need to revert it would be easier for them to revert to the previous version. • New versions should be approved by all developers and the manager before being executed and to be checked that it is of quality and compliance after being made

,eg: bug-free to prevent minimize application downtime , security management is done to prevent hackers from hacking

- Use a configuration management system and make processes progressively
- Maintenance should be done regularly to avoid having any bugs/loopholes where any unforeseen users having access to data that they are not suppose to have access to it

Don’t underestimate anything we have done and must always be careful and precautious to obtain only the correct and accurate estimation for all the time and effort on integration, testing, documentation and maintenance of the application.

Code review is a very effective and efficient method to find any software defects for the application. Planning and managing code reviews between team members is a necessity as it helps to ensure that codes are simple and hence reduction of application loading time as lesser code will make the application run quicker.

Software testing will use both black box and white box testing. It will involve

1) unit testing,

2) functional testing,

3) integrating testing

4) acceptance testing.

8. Risk Management

Besides the general risk management, the following risks shown below have been identified for the HouseHunt application project:

Specification Delays

Impact Severity: High

Probability: 15%

Impacts: Delaying finalization specification from customers will definitely push our schedule for all following stages of the project.

Risk Reduction: We must monitor the progress of specification carefully and always meet the deadline and verify the specifications with the customers from time to time.

New Additional Requirements are being added during development

Impact Severity: Medium

Probability: 15%

Impacts: Depending on the progress we had made for the application so far, if the new additional requirements does not affect our development as it is a dependent new feature, it would be easy to implement it as it would not affect what we had done early, However, if the new additional requirement will affect the current coding, more time will be wasted as we might need to redo the entire code from scratch due to this new additional requirement requested by the customer

Risk Reduction: We must always be rigorous in eliciting requirements. We must make sure that our customers are aware of how and what we are using to create this application for example Django, SQL in order to prevent them from requesting a change for different software to use to implement the creation of the application during the development phase. We must also make sure to specify all the different requirement being requested by the customer in our proposal for confirmation to be made so that all functionality the customer specified and required are being documented before the start of the application development

Application Handover Delay

Impact Severity: High

Probability: 25%

Impacts: Delay in handover of the application to the customer should be minimized as much as possible. If we fail to meet the required deadline, we might end up losing the customer's trust and it will impact our company reputation.

Risk Reduction: We must always work towards having a full test run 1 week before the handover to avoid any implications. We must also monitor the progress of our application creation carefully. Be sure to follow and work according to the project schedule planned in ensuring delay will not happen. Seek help from others if needed, request for additional colleagues to help out the project if any are available.

Application System Memory size being Underestimated

Impact Severity: Moderate

Probability: 20%

Impacts: It will cause the application to result in having a downtime being unable to be used by the customer.

Risk Reduction: We must always check on the system memory size being used up during development and ensure that it is at least 15-20% of memory free based on the maximum data storage requested by the customer system.

Problems coordinating within group

Impact Severity: Moderate

Probability: 40%

Impacts: Members may be unaware of what is expected of them; managers may not be able to measure progress; portions of projects not completed.

Risk Reduction: Follow communication plans as documented

Customer cancels project

Impact Severity: High

Probability: 1%

Impacts: All work done will be wasted.

Risk Reduction: We must keep in good relationship and close contact with our customers at all times. Ensure that we have done some market research indicating a demand for this product for example having a progress payment on every development phase. Hence they will have to progressively pay us the application during the process where we are creating the application. For example, customer have to pay us 50% during the start of development phase 30% during 2nd development phase and remaining 20% during the final phase However we must always take note to listen honestly to their feedback to resolve any of their concerns

9. Quality Assurance

The project will achieve quality assurance by following the standard set by the company. The specification procedures and details shall be provided in the Quality Plan. Specific test procedures and details shall be provided in the System Test Plan.

We will be using **Unit Testing** to test our system components individually follow by **In-Place Testing** whereby we test the whole system as a single unit.

Furthermore, **System Function** will be tested to ensure that all software flaws are being found and fix as soon as many are being found in ensuring a tight security management for the application to prevent any unwanted guest from accessing the data that they shouldn't have rights to.

10. Monitoring & Control

We applied many different procedures in order to be able to successfully monitor and control the progress of our software application project. We make use of

Proper allocation of resource management: Estimation of the HouseHunt's resource requirements, depends primarily on team resources, which includes providing an allocation of project progress when compared to the progress in terms of the project development phase. The allocated estimation of each development resource requirements provided in this document will allow us to easily track the progress of the project and determine whether more resources are needed for the project.

Identifying all possible major project risks: Majority of the risks we anticipated have all been identified in our Risk Management section of this document, along with the measures that we should all take in order to avoid them if any of them happens we will follow the risk reduction we written to reduce having more issue face helps us as it helps us to consider our immediate response activities in light of existing and new disastrous risks that might happen later on.

Weekly updates and reviews of members on project progress: Throughout the entire process of HouseHunt project creation, all project members shall at least provide a weekly update on their progress so that it can be reviewed by all members on the progress of their allocated project tasks, including management, planning, analysis, development, and testing of codes or documentation.

Planning And Task Decomposition: This documentation documents an estimated timeframe for the task to achieve and project plan to be completed. We provide a reasonably achievable and accurate timeline whereby all members are able to achieve in completing the task we gave to them by splitting the tasks into different subcomponents and estimating the requirements needed for each task. Throughout the implementation phase, these subcomponents can allow us to measure the progress of our project. Project subtasks and timeframe estimates are all documented down in the Estimates and Work Breakdown Structure sections of this documentation.