

RoomieLah Project

Finding roommates? Easy!

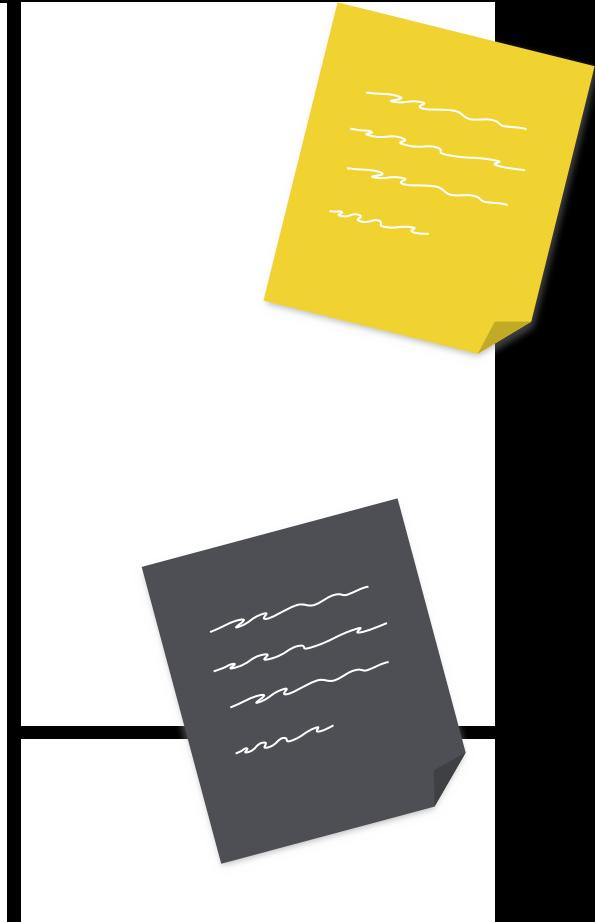


Table of Contents

01

Our Team

03

**Design for
Maintainability**

05

Project Management

02

Product Introduction

04

**Software Quality
Assurance**

06

Risk Management



OUR TEAM





Srishti Arora
Project Manager



Aks Tayal
QA Engineer



Pratyush Pandey
Lead Developer



Atul Acharya
QA Manager



Gopal Agarwal
Release Manager



Sanath Surawar
Front-end Developer



Rajagopal Iyer
Back-end Developer

Why RoomieLah

Big issue among university students is finding a good roommate.

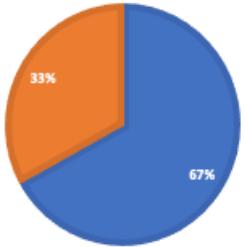
App facilitates matching with a potential roommate.

Helps effectuate a better campus experience since roommates are crucial.



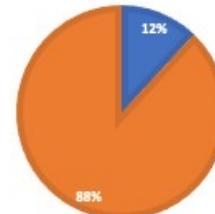
DO YOU LIVE IN A SHARED ACCOMODATION?

■ Yes ■ No



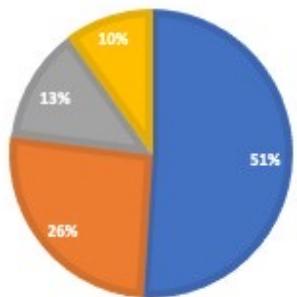
WERE YOU ABLE TO FIND SUITABLE ROOMMATES?

■ Yes ■ No

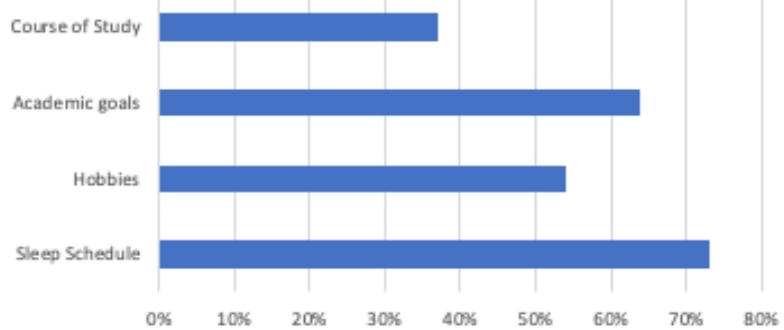


WHAT IS YOUR STUDY YEAR?

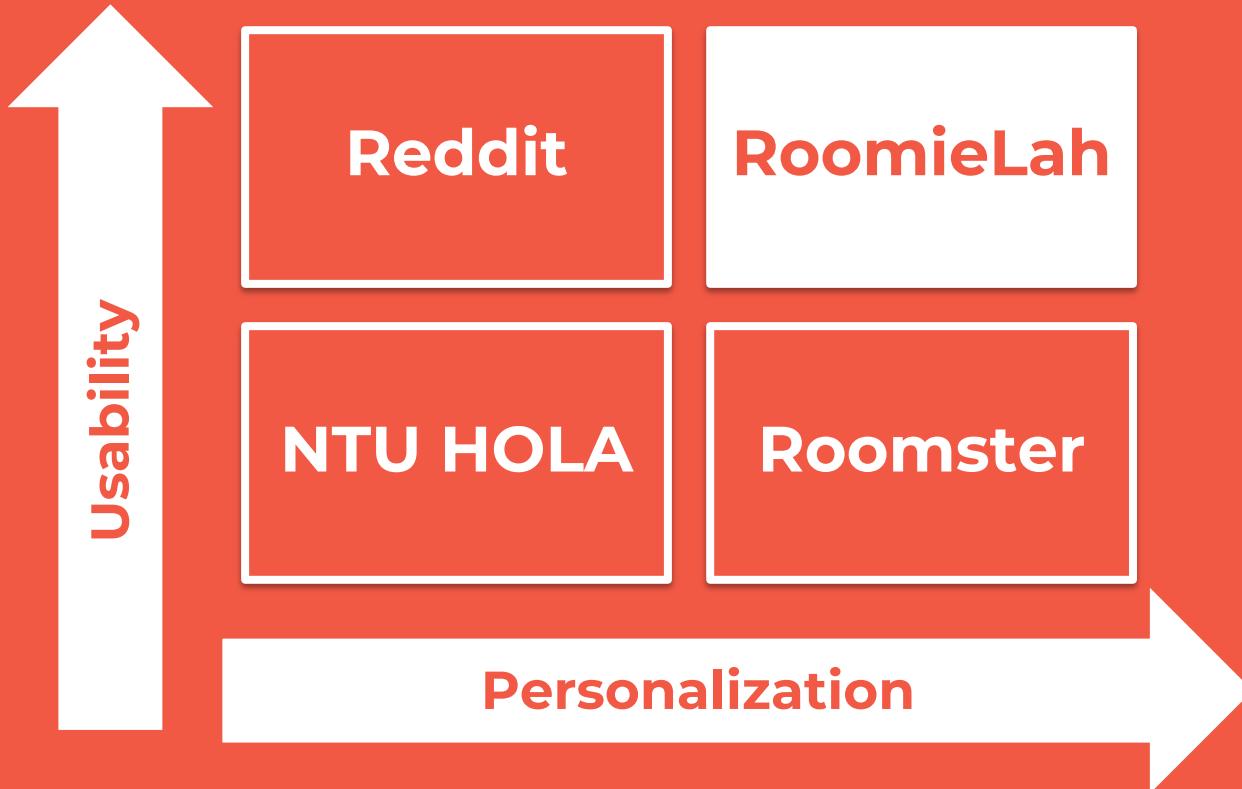
■ 1 ■ 2 ■ 3 ■ 4



Which of the following qualities are most important when selecting roommates?



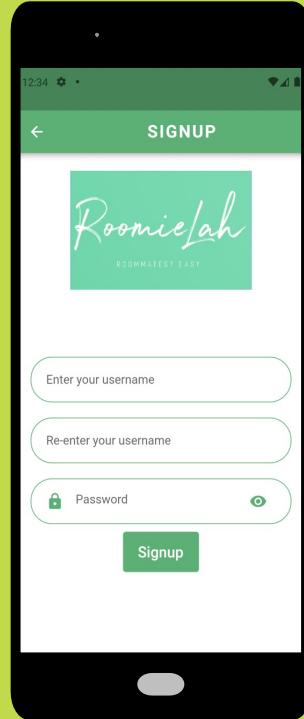
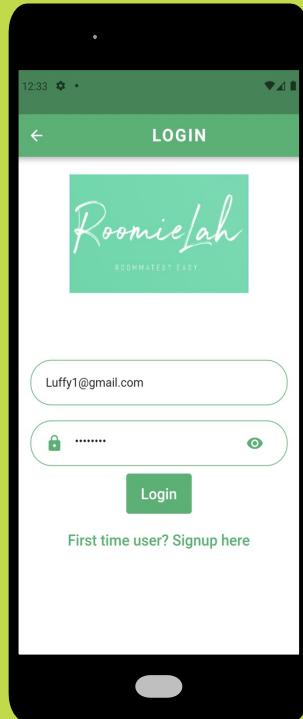
Competitor Analysis



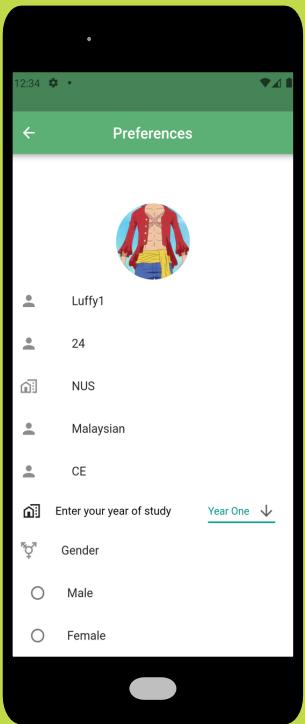
SCREENS



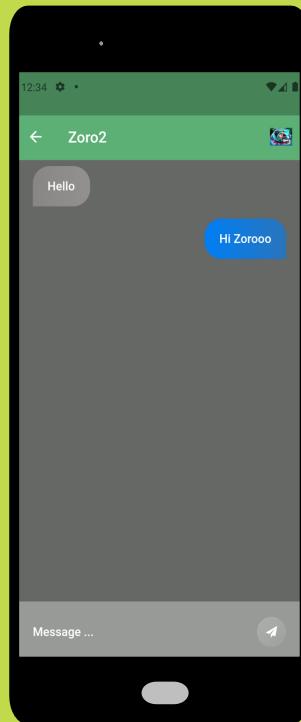
Login & Sign Up



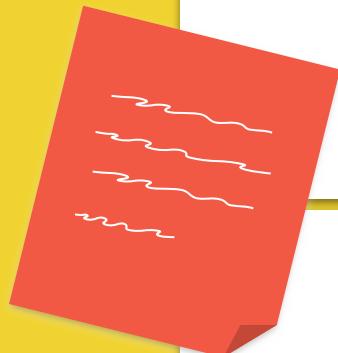
Home and Profile



Chat Screens



Design for Maintainability

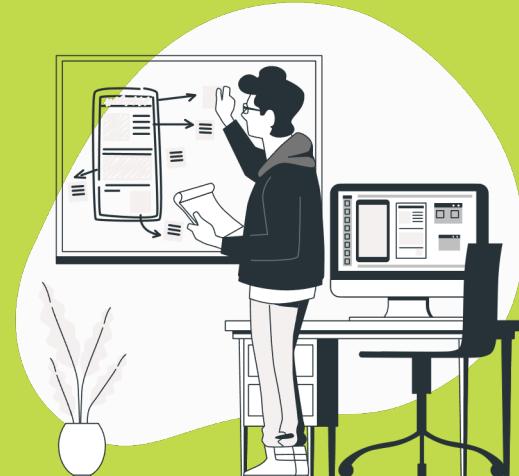


03

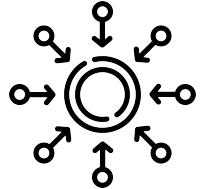


1. Architecture considered for easy maintainability.
2. New features constantly added since developer team had first-hand experience with the problem statement.
3. Design was implemented by effecting a:
 - Requirement Elicitation Process
 - Competitor Analyses

Early Design



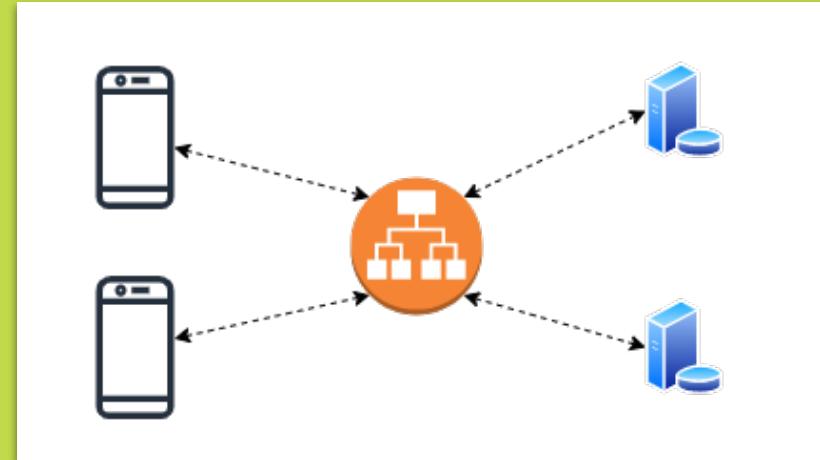
System Design



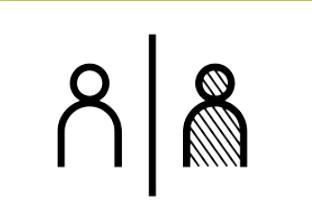
Centralized



Scalable

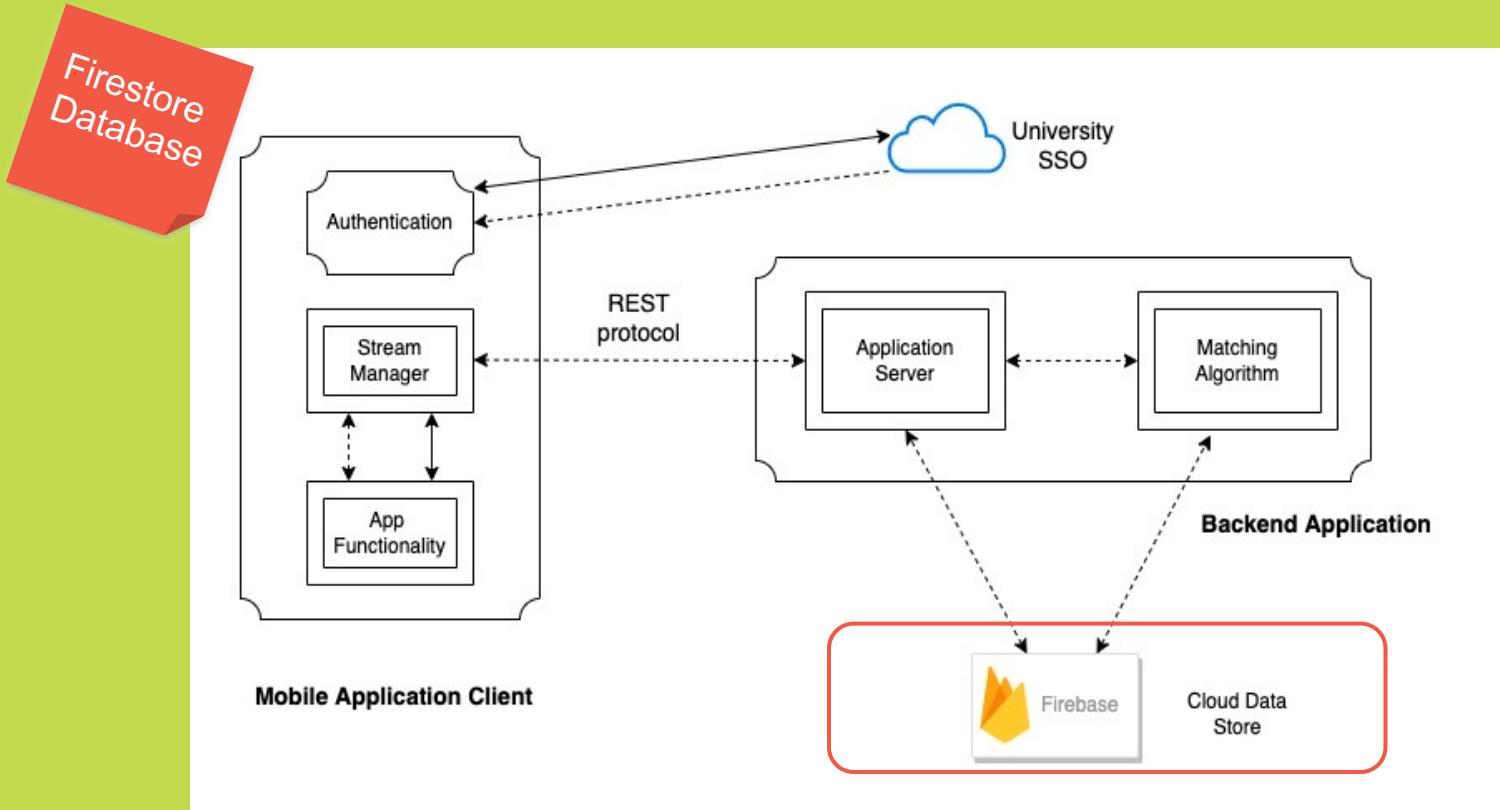


Client-Server Architecture

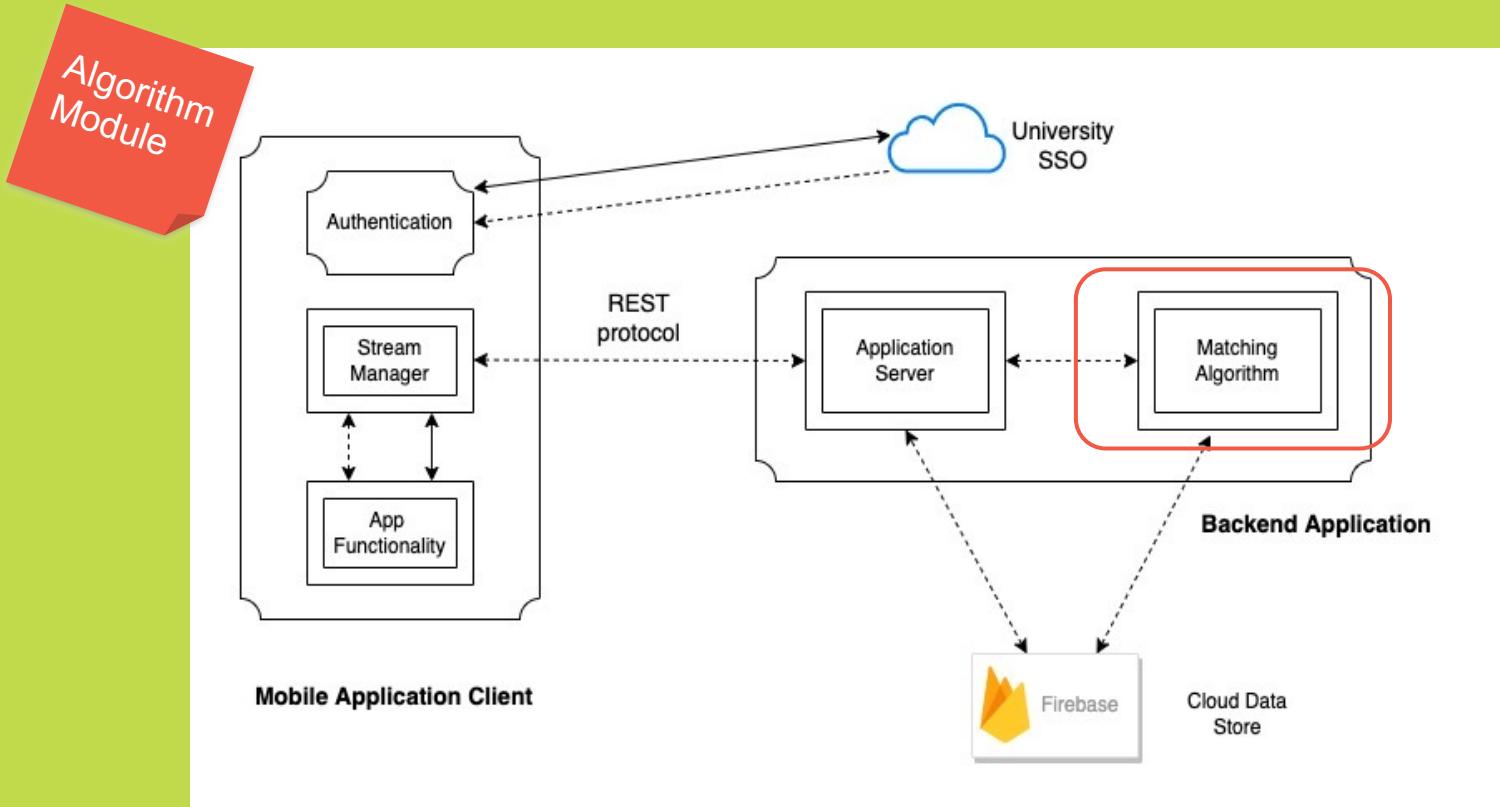


Segregated
Modules

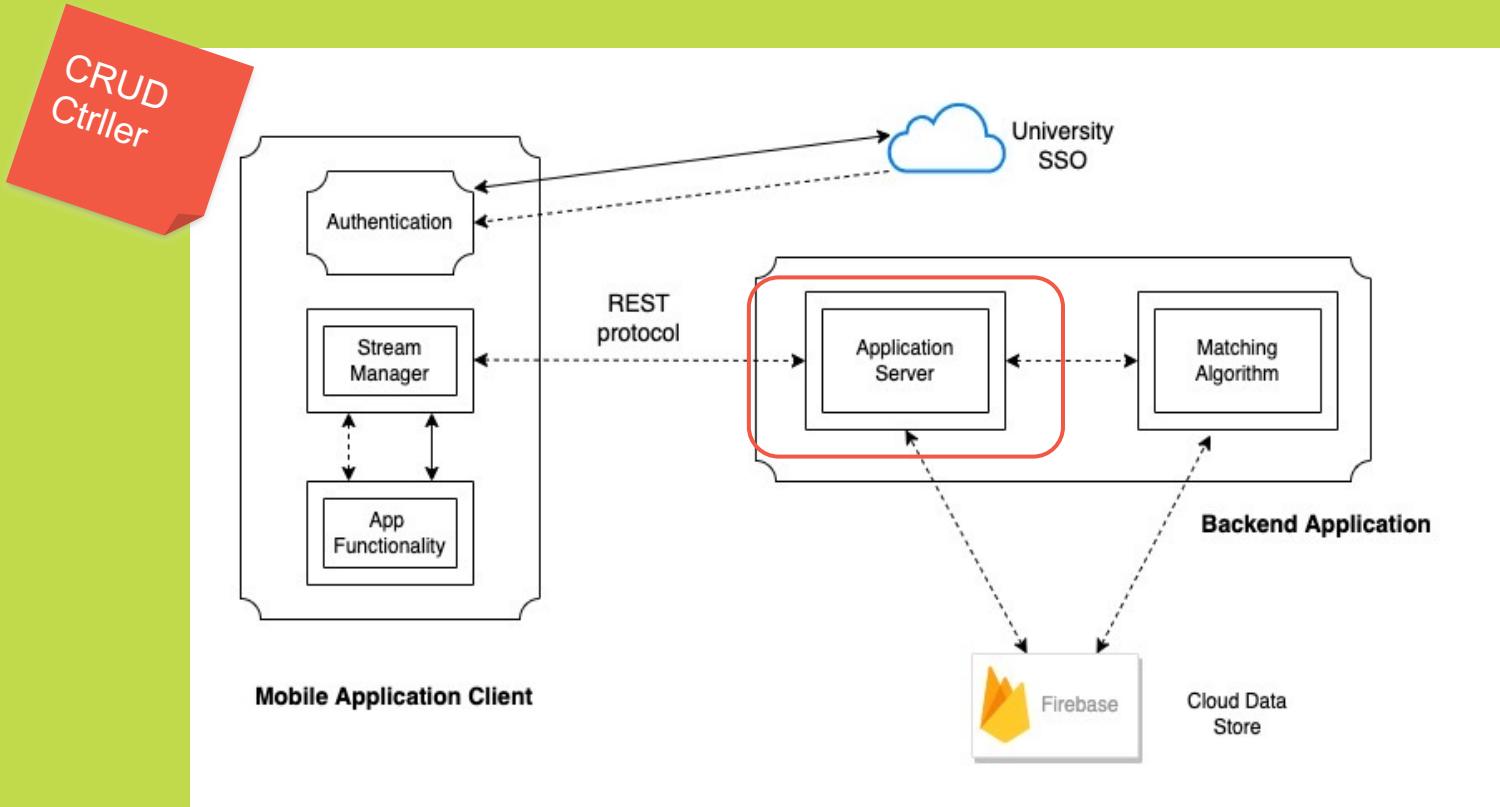
System Design



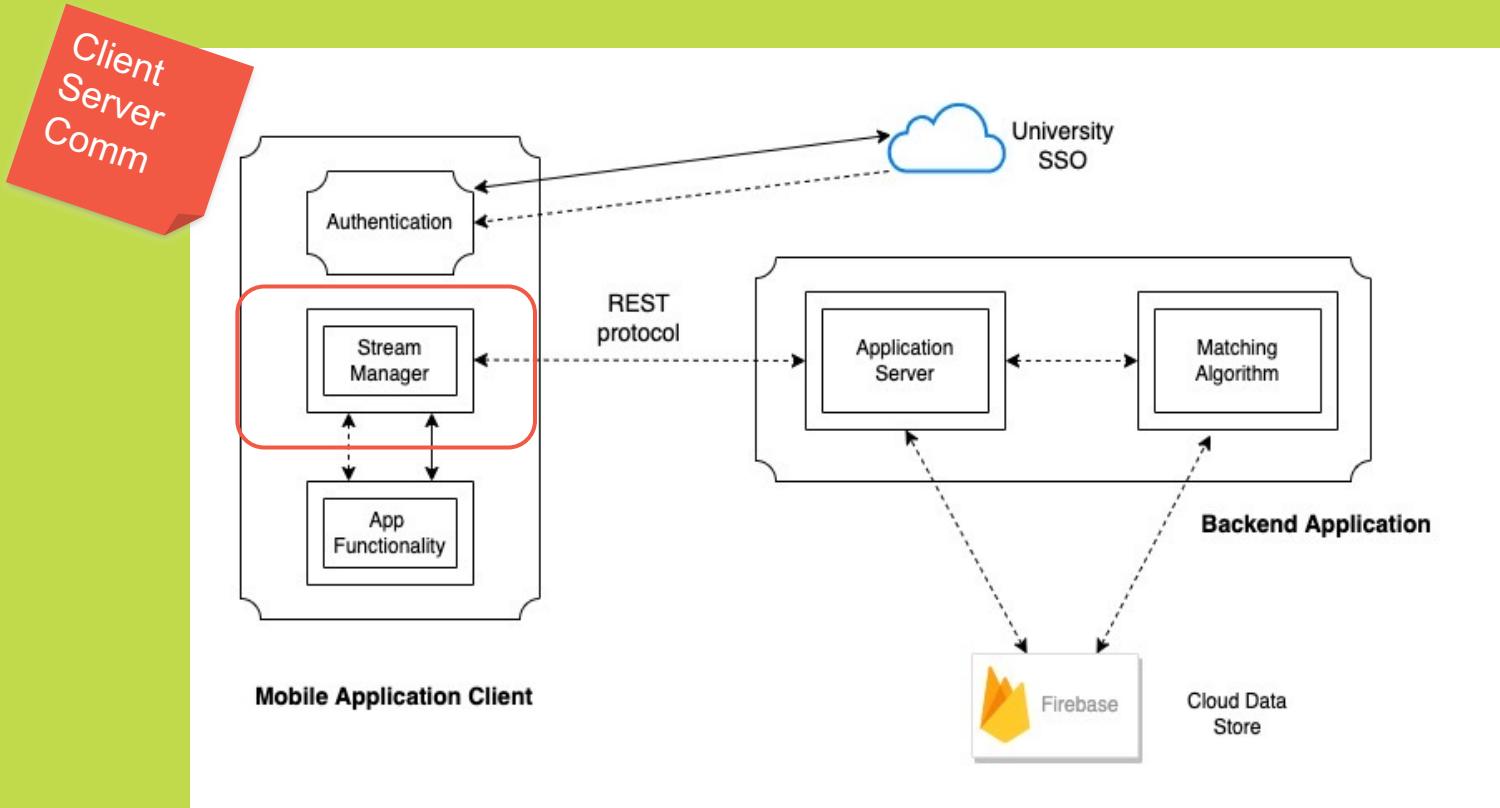
System Design



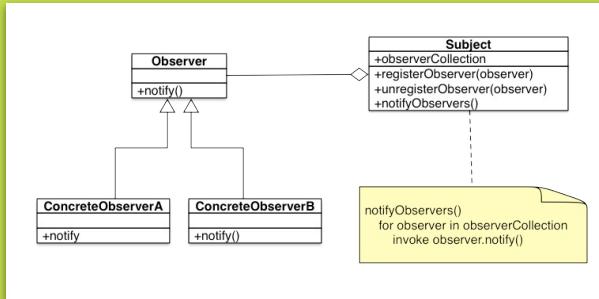
System Design



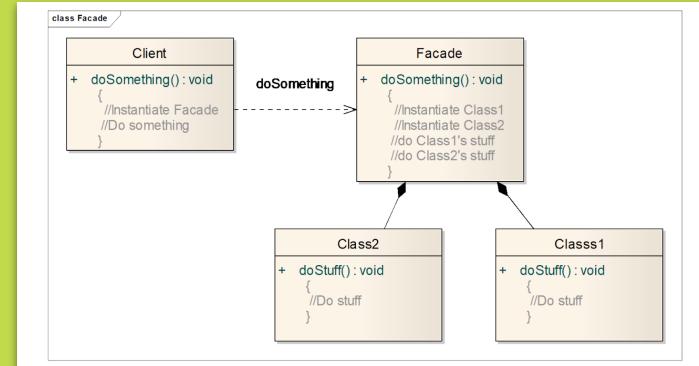
System Design



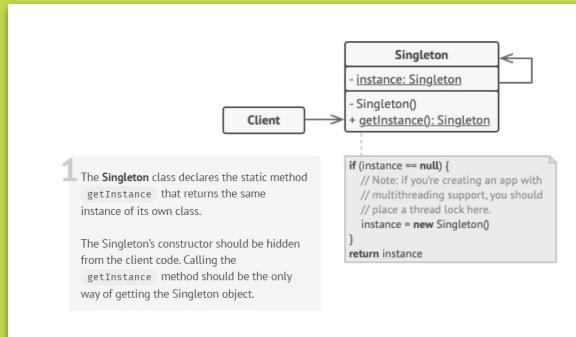
Design Patterns



Observer

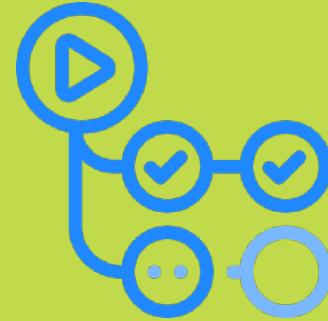


Facade



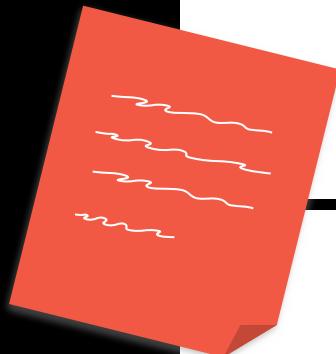
Singleton

CONFIGURATION MANAGEMENT



Software Quality Assurance

04



Quality Criteria

Usability

The level of ease with which the system allows users to reach their goals

Maintainability

Software's ability to maintain itself over long periods of time

Scalability

Measure of system's ability to increase or decrease performance based on system load

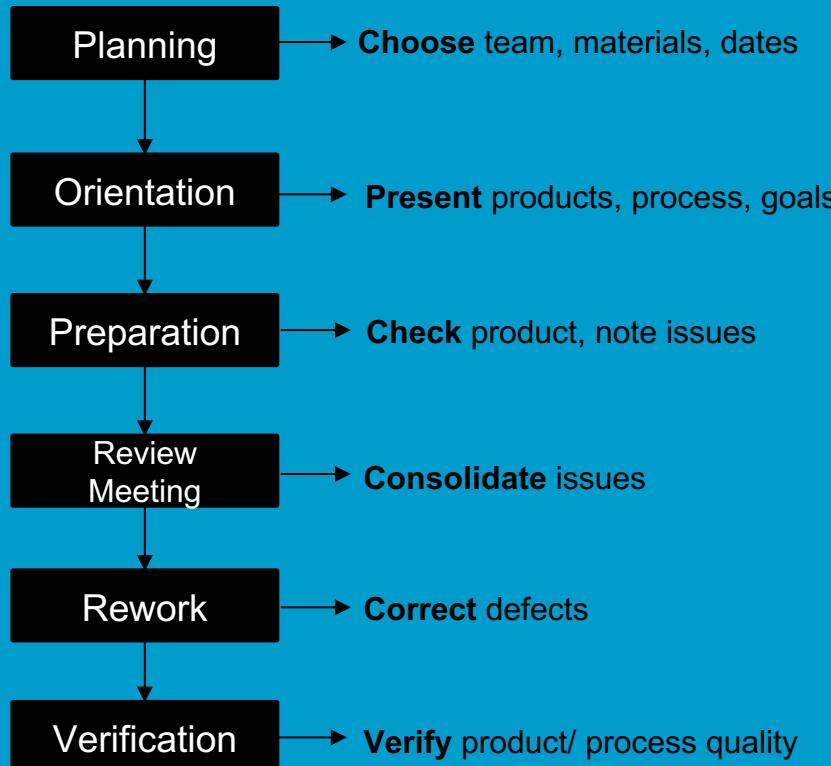
Functional Suitability

Functional completeness & correctness

Portability

Ability to move software to a different platform

Technical Review



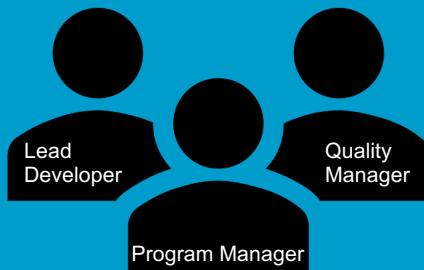
Walkthroughs

- Peer review was lead by Lead Dev
- Participants ask questions and make comments about possible errors
- Iterative rectification of issues identified

Inspections

- Control technique to ensure that the documentation produced in a given phase is consistent with that of previous phases
- Formal review

Management Review



Management Review Committee



Review Meeting Schedule



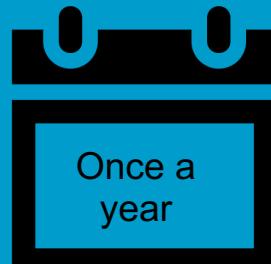
Evaluate the impact of actions and ways to measure those impacts

Audits

Auditors who are **not members** of RoomieLah development team, examine the quality independently.

They assess the quality of:

1. Software product
2. Software processes
3. Compliance with specifications



Software Audit Schedule



Process Goals

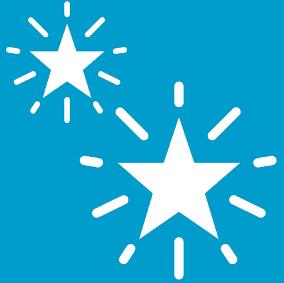


Identify key areas of quality improvement

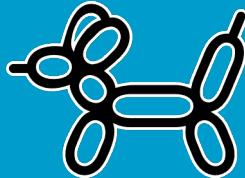
Establish a quality culture in the team

Traceability in product lifecycle, technical soundness etc. for RoomieLah

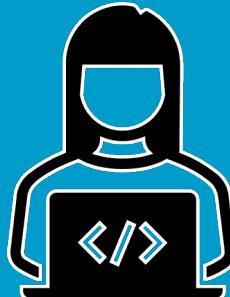
Commitment to Perform



Senior management provides sponsorship and bonus to introduce new tools and technologies which ensure high quality of the product.

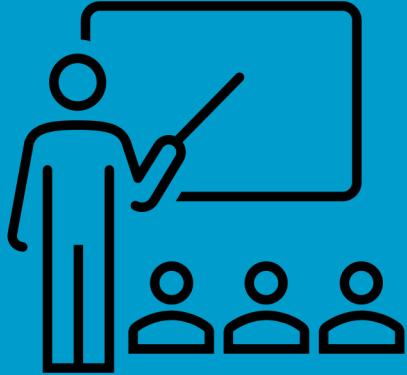


Briefings on project metrics and organization policies delivered through interactive activities



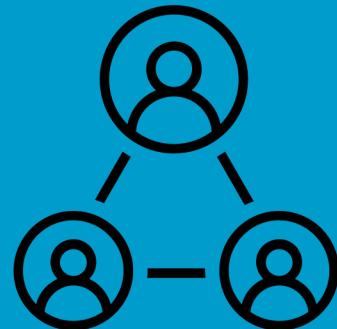
QAM to gauge commitment from team and conformance to proper standards

Ability to Perform



Knowledge Enhancement

- CZ3002 Lectures
- CZ3002 Tutorials

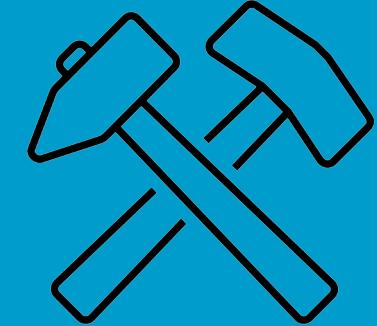


Resources

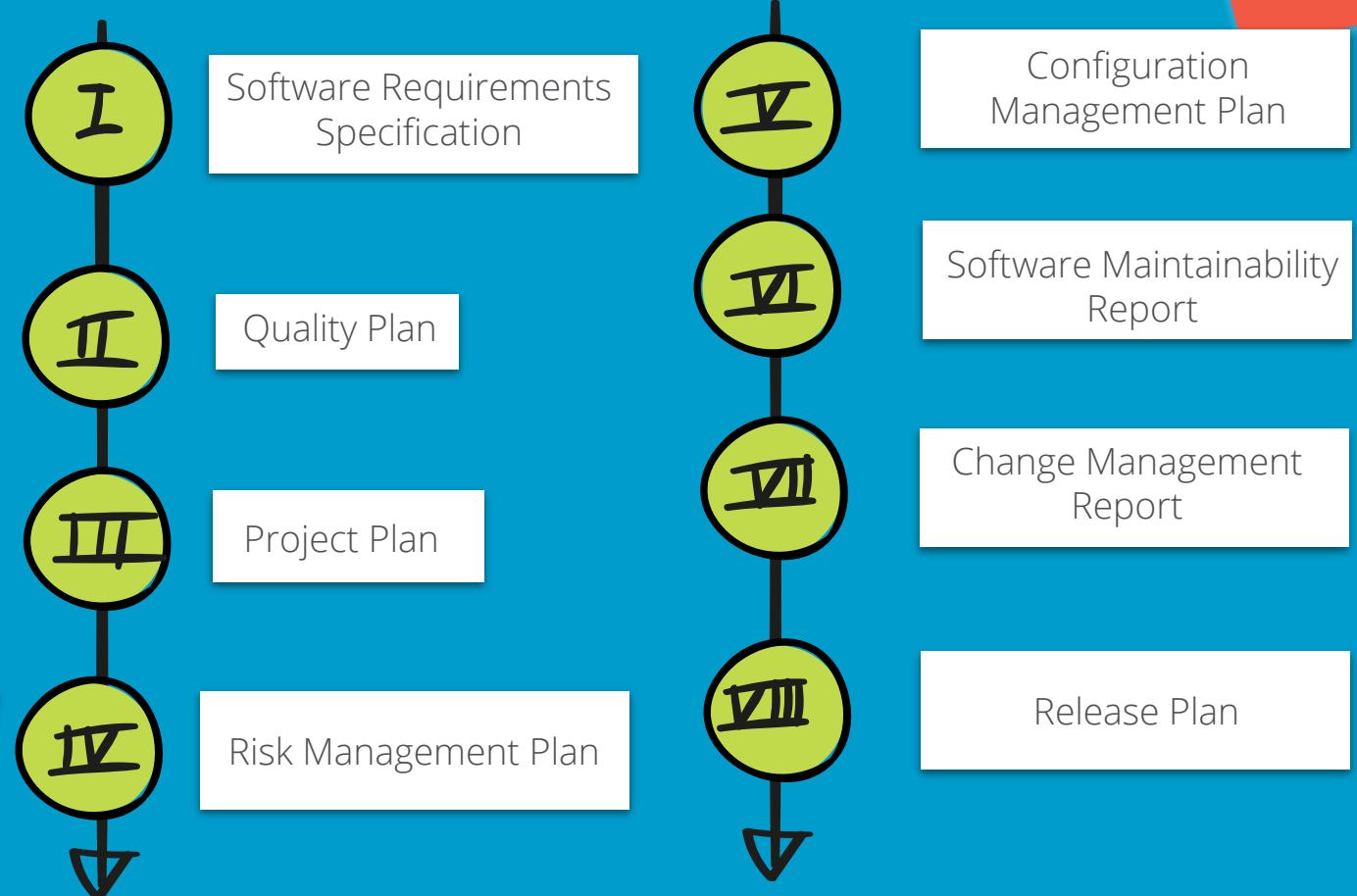
- Access to lab servers, PC
- Cloud database

Tools

- GitHub VCS
- VSCode & XTools



Activities Performed



Working & Tracking

- Collaborative implementation through GitHub Version control System
- Issues created in GitHub to track the work
- Corrective action for any errors or bugs is taken through a Change Request Workflow

A screenshot of a GitHub repository interface. At the top, there is a search bar with the query "is:issue is:open". Below the search bar are several navigation and filtering options: "Filters", "Labels 9", "Milestones 0", and a green "New issue" button. A dropdown menu shows two open issues: "#2 Create chat UI and connect to Firestore" and "#1 Create UI screens for login and signup". Both issues are labeled "enhancement" and "good first issue". Issue #2 was opened 16 days ago by Pratyush0411, and issue #1 was opened on 14 Feb by tayalaks2001. There are also icons for a profile picture and a comment count of 1.

Issue #	Title	Type	Author	Opened On	Comments
#2	Create chat UI and connect to Firestore	enhancement	Pratyush0411	16 days ago	1
#1	Create UI screens for login and signup	enhancement	tayalaks2001	14 Feb	0

Measurement, Analysis & Verification

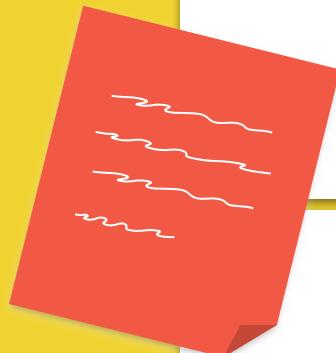
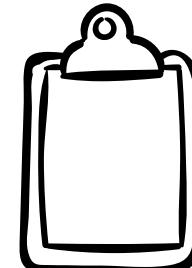
Periodic assessment and reporting of the project's progress throughout the SDLC

- The performance metrics are made quantifiable and verifiable
- The results analysed by PM , and changes suggested based on the current state
- The amendments verified using audits by management and SQA

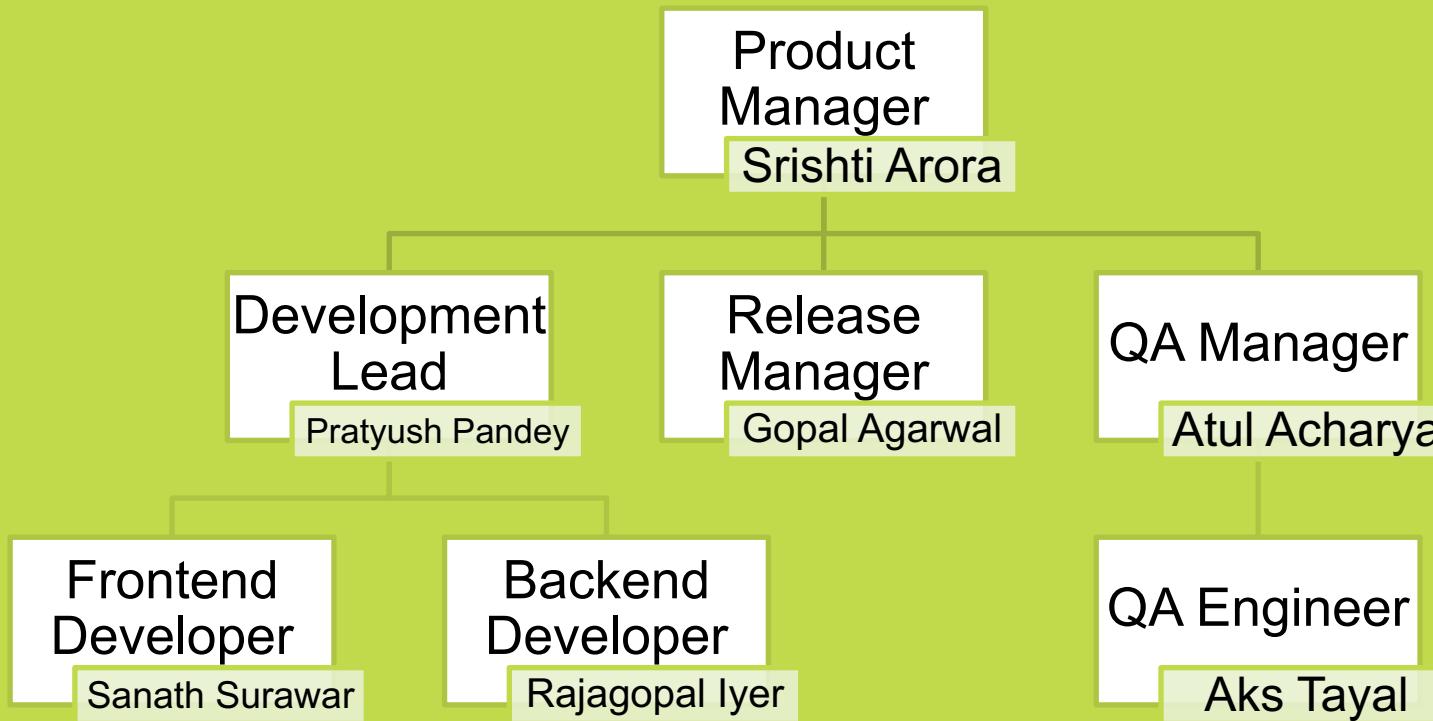


Project Management

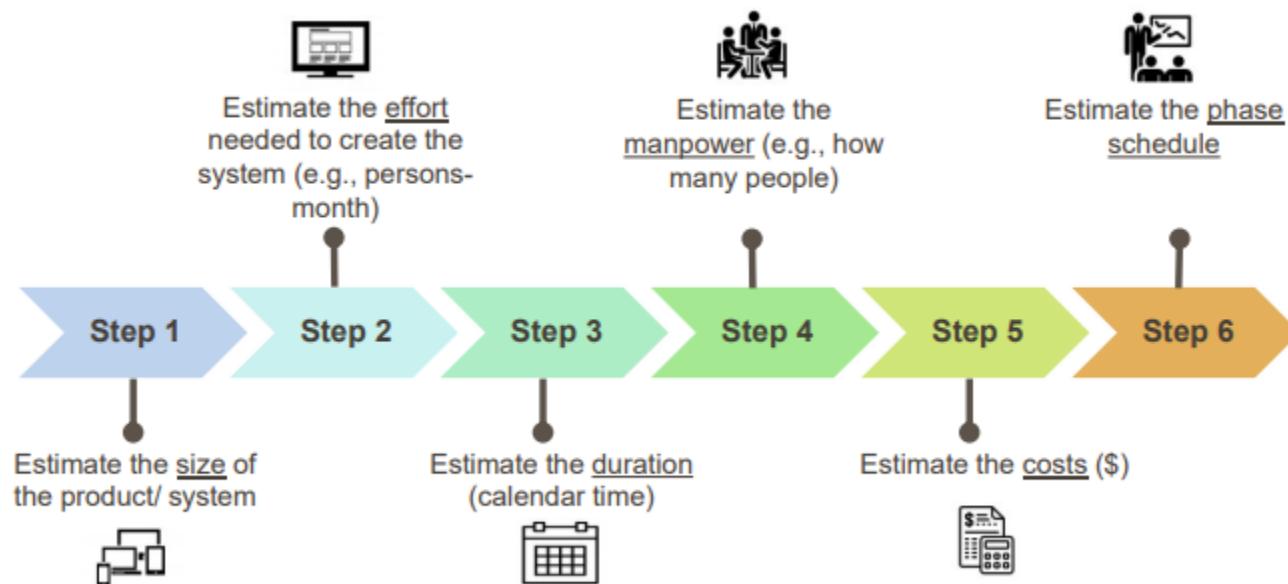
05



Project Organization



Project Estimate



Function Points (1/2)

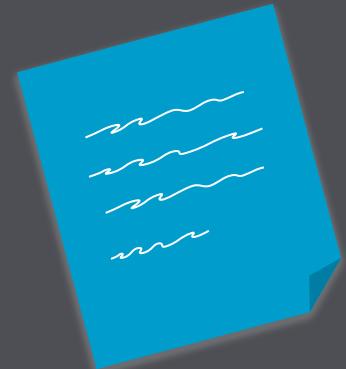
Characteristic	Complexity	Name
Inputs	Low	Registration
		Login
		Chat Room
		Edit Profile
Outputs	Low	Viewing Matched Profiles
	High	Viewing Recommended Profiles

Function Points (2/2)

External Interfaces	Medium	Chat Room
		User Login
		User Registration
		Edit profile
		(Cloud Backend)
Logical Files	N/A	
Inquiries	N/A	

Unadjusted Function Points

Total = Low + Medium + High = 71



Adjusted Function Point

Influence Factors = 7
(Data Comms,
Complex Processing)

Total Influence Score
= 20

Adjusted FP =
Unadjusted FP ×
Influence Multiplier
• = $(71 \times 0.85) = 60.35$

Influence Multiplier =
 $(20 \times 0.01) + 0.65 =$
0.85

Lines of Code



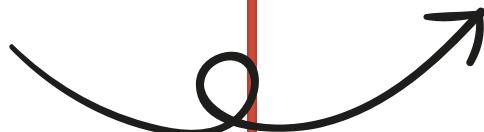
47.41 LOC/FP

Capers Jones Statistics
for application
implemented using Dart



2861 LOC

Adjusted Functional
Point x LOC/FP



DISTRIBUTION OF EFFORT

1990's Industry data	Work Package	Distribution	Estimates (PH)
Preliminary Design (16%)	Project Plan	5%	29.34
	Requirement Specification	9%	52.82
Detailed Design (22%)	User Interface	7%	41.08
	Technical Architecture	15%	88.03
Code & Unit Testing (26%)	Data Modeling	9%	52.81
	Implementation & Unit testing	21%	123.24
	Online Documentation	5%	29.34

DISTRIBUTION OF EFFORT

Work Package	Distribution	Estimates (PH)
Integration and Quality Assurance	29%	170.20
	Extrapolated total effort	586.88
	2% for project management	11.73
	3% for contingency	17.61
	Total effort	616.2

COST ESTIMATE (1/2)

Item	Cost Estimate
1 - iPhone SE (to test RoomieLah on iOS)	\$ 649
	\$ 500
1 - Google Pixel 6 (to test RoomieLah on Android)	
Single Core	
1 GHz	
4 GB Ram	
Total	\$ 1149

COST ESTIMATE (2/2)

SOFTWARE COSTS

Microsoft Office 2000	\$0.00
Android Studio	\$0.00
Visual Studio Code	\$0.00

EMPLOYEE COSTS

7 Employees with
568.88 working hours
with \$18.00/hour

\$10,239.84

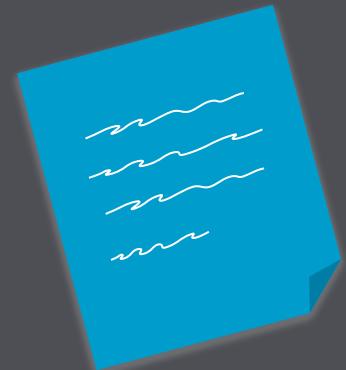
MISCELLANEOUS COSTS

Paper, notebooks,
photocopying and
other costs

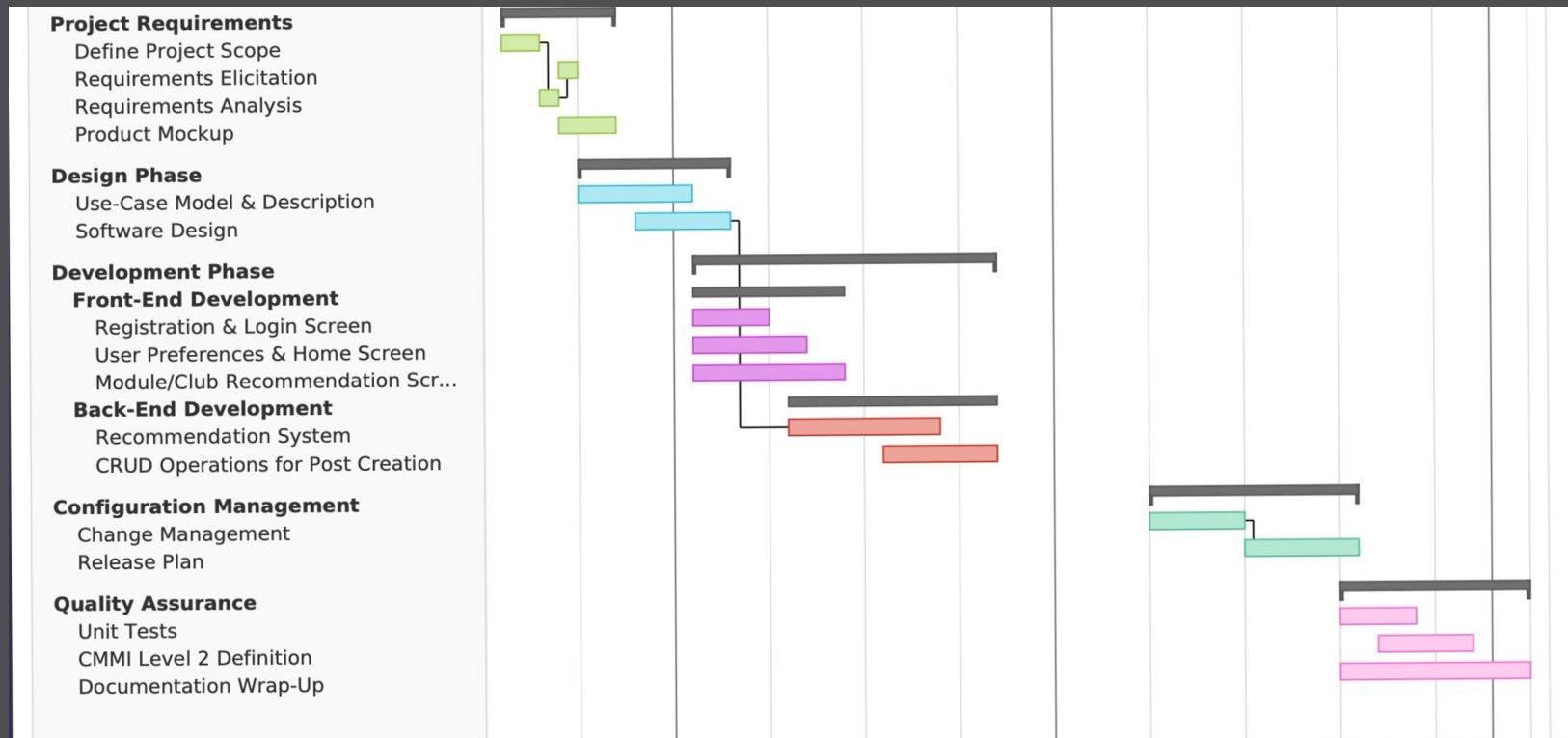
\$50

TOTAL COST

11,438 SGD

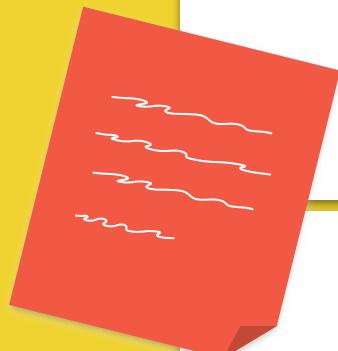


GANTT CHART

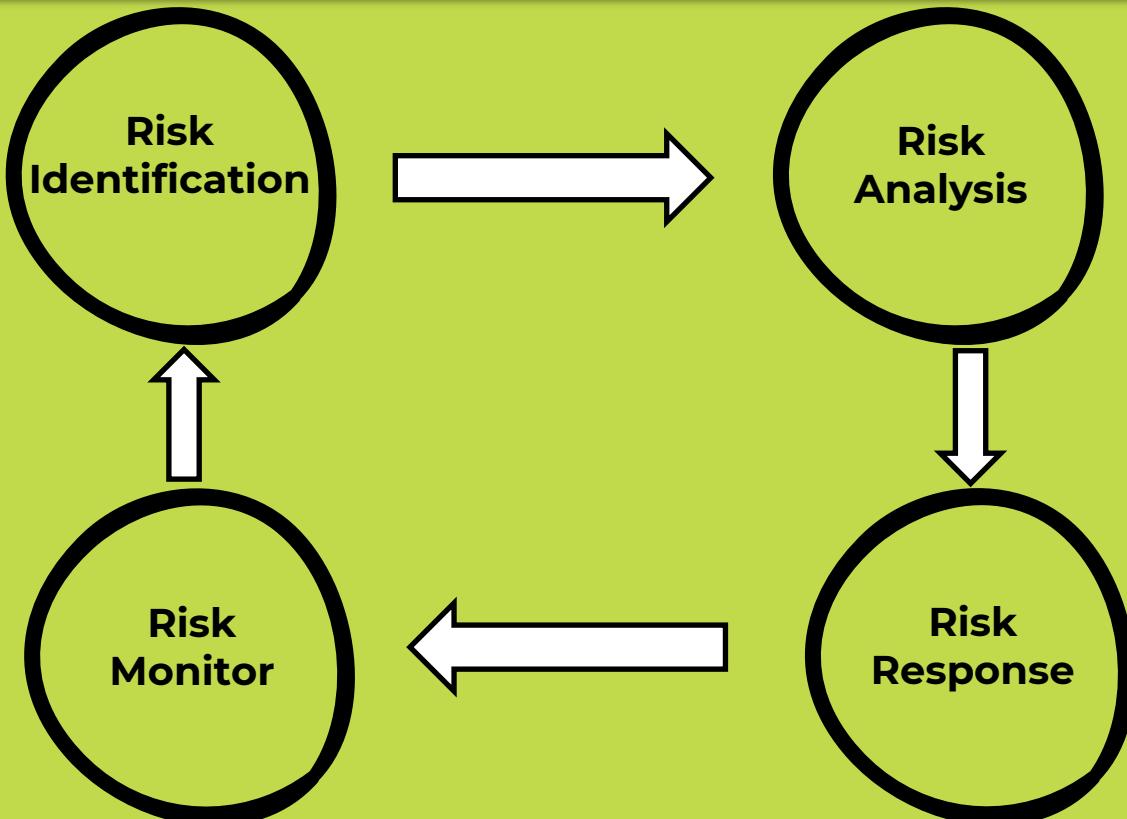


Risk Management

06



Risk Management Process



Types of Risks Identified

Requirements
elicitation

Technology

Budget

People

Estimation

Tools

Organizational

Risk #1

Technology

The system might not work as intended on the end devices due to device requirements.

The server potentially crashes, and essential data is lost.

The app might face performance issues due to malfunctions of Firebase APIs.

The Version Control System in use might experience fallacies.

Risk #2 People

The key people working on the project might leave the project.

The people working on the project might fall ill and would not be able to work for a period of time.

The developer team experiences domain knowledge inadequacies.

The team lacks motivation to complete deliverables.

Risk #3 Estimation

Time taken to finish a deliverable is underestimated

Amount of resources required to finish a deliverable is underestimated

Risk #4 Organizational

The organization might be restructured causing a change in the people responsible for different components that might lead to confusion.

The project might introduce contract-based members, who can affect team communication and morale.

Risk Analysis



QUALITATIVE



QUANTITATIVE

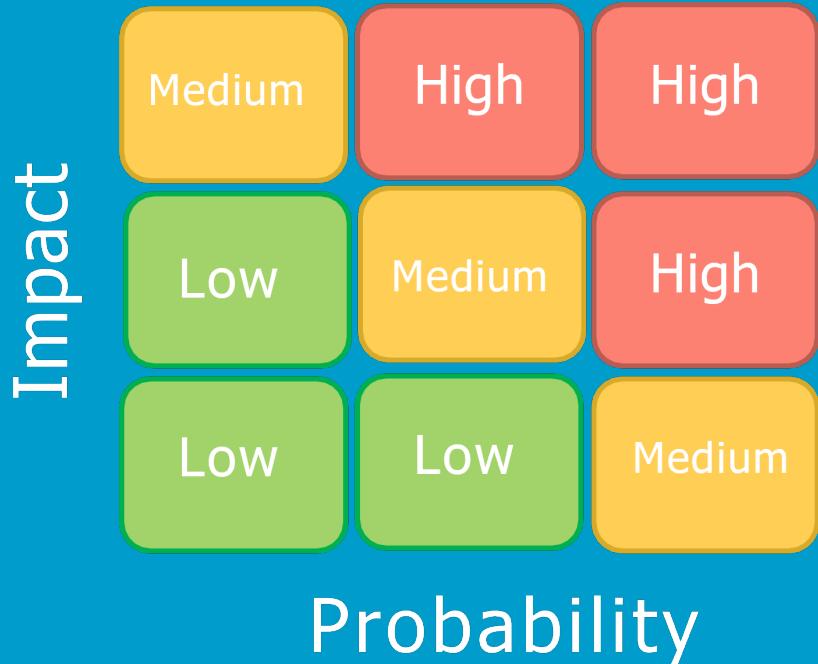
Qualitative Risk Analysis

Probability of occurrence of risk

- **High:** $P > 70\%$
- **Medium:** $P = 30 - 70\%$
- **Low:** $P < 30\%$

Impact of risk on project:

- **High:** great impact
- **Medium:** slight impact
- **Low:** little impact



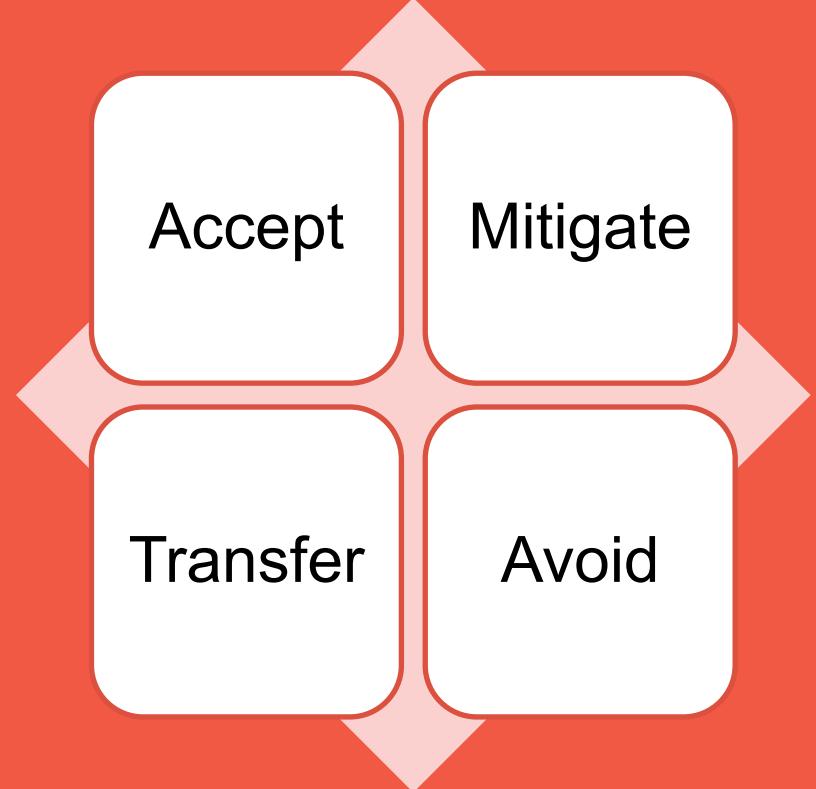
Impact – Probability Matrix

Impact		High	Probability		
			Low	Medium	High
	High	<ul style="list-style-type: none">1. Server Crash2. Non-scalable deployment servers3. Device requirement not fulfilled4. Inefficient Code5. Underestimation of Project Size	<ul style="list-style-type: none">1. Reused Software Component Issues2. High Rate of bugs & issues3. Version Control fallacies.		
	Medium	<ul style="list-style-type: none">1. Database Limits2. API Malfunction3. Developer Management Conflict4. Undermined Motivation5. Project Restructure6. Inadequate Domain Knowledge7. Abandonment in high pressure8. Under/Over Estimation of user base	<ul style="list-style-type: none">1. Requirement Changes2. New use cases3. Time Underestimation4. Client miscommunication5. Team member illness6. Single point of failure (e.g., data loss) affecting project timeline7. Difference in Functional Requirements documented and deployed.		
	Low	<ul style="list-style-type: none">1. Project Management Restructure2. Contract-based employee affecting team comms.	<ul style="list-style-type: none">1. Dorm housing structural revamp.		

Quantitative Risk Analysis

Area of Risk	Severity	Likelihood	Level of Control	Significance
Technology	3	2	2	7
People	3	2	2	7
Organizational	2	1	2	5
Tools	2	1	2	5
Requirements Elicitation	3	3	1	7
Budget	3	3	1	7
Estimation	2	3	2	7

How do you respond to a Risk?



Accept

Organizational

Organizational changes must be accepted and necessary changes must be made.

Requirements Elicitation

Changes in the requirements must be accommodated and necessary changes must be made to the project timeline

Budget

Small changes in the budget is expected and must be accepted

Mitigate

People

Ensure the overall cohesion of the team using team bonding activities

Tools

The capabilities as well as the shortcomings of the tool must be identified to ensure that it satisfies the requirement without any issue.

Estimation

Sound logical consensus of the team must be present before any estimating any part of the project.

Avoid

Technology

- Identify the most appropriate technology stacks such that it provides robustness
- Invest in highly reliable and scalable database, cloud service and other such service.

Risk Monitoring

Keep track of the identified risks

Be on the lookout for new risks

THANK YOU

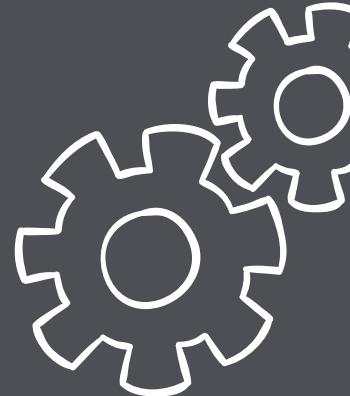
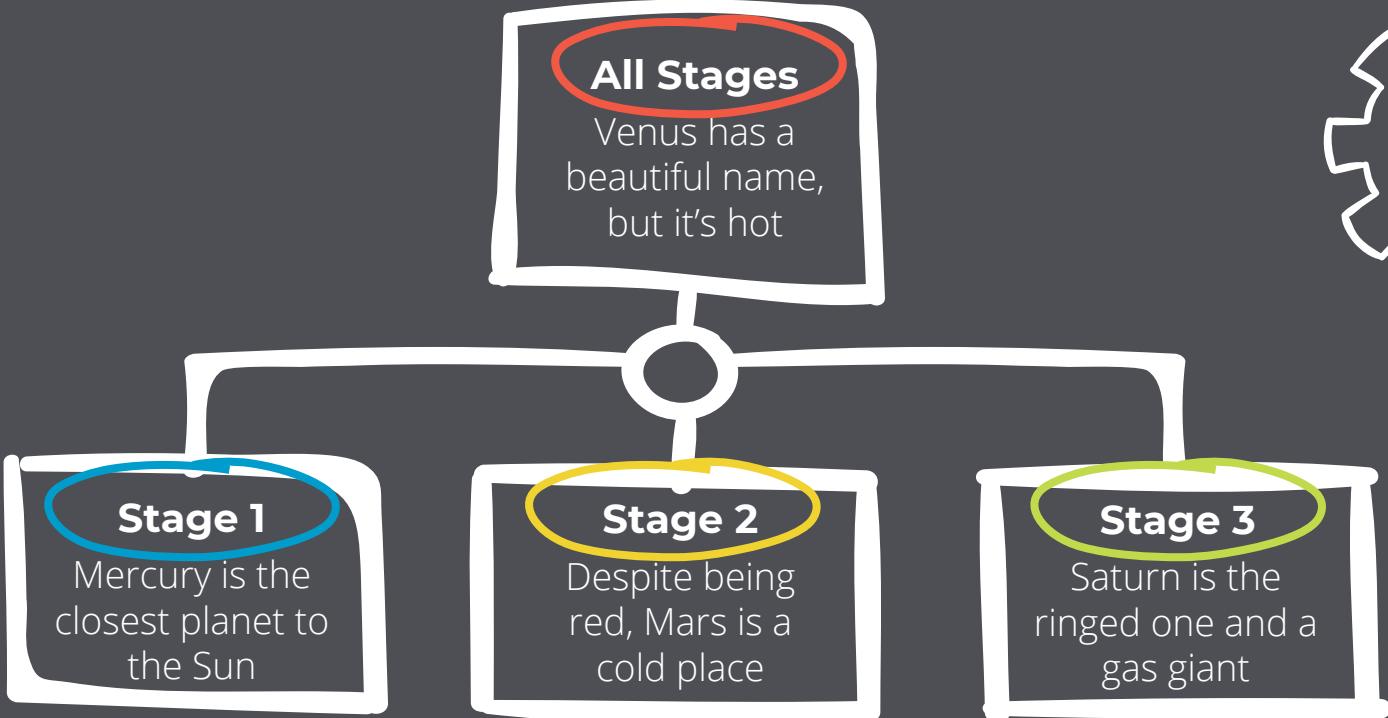
Any Questions?

Sneak Peek

You can replace the image on the screen with your own multimedia content. Just delete this one, add yours and center it properly



Project Stages



Timeline

Venus

Venus has a beautiful name, but it's hot

Jupiter

It's the biggest planet in the Solar System



Mercury

Mercury is the closest planet to the Sun

Mars

Despite being red, Mars is a cold place

Our Partners



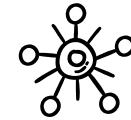
Venus

Venus has a beautiful name, but it's hot



Mars

Despite being red, Mars is a cold place



Saturn

Saturn is the ringed one and a gas giant

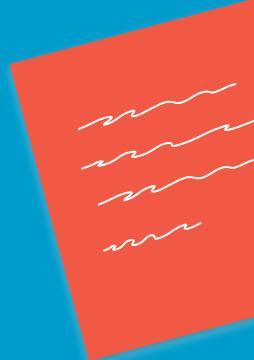
Our Team

James Smith

Here you can talk a bit about this person

Kevin Smith

Here you can talk a bit about this person



Alternative Resources

