



NTUCollab

A Collaboratory Platform for NTU Students

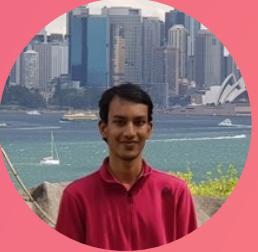
By Team Eagles

Team

Dandapath Soham
PROJECT MANAGER



Somani Palak
QA MANAGER



Bansal Aditya
QA ENGINEER



Kanodia Ritwik
RELEASE MANAGER



Gupta Jay
DEVELOPMENT LEAD



Mundhra Divyesh
FRONT-END DEVELOPER



Bhatia Ritik
BACK-END DEVELOPER





Outline

- 1 Product Introduction
- 2 Design for Maintainability
- 3 Software Quality Assurance
- 4 Product Management
- 5 Risk Management

Product Introduction

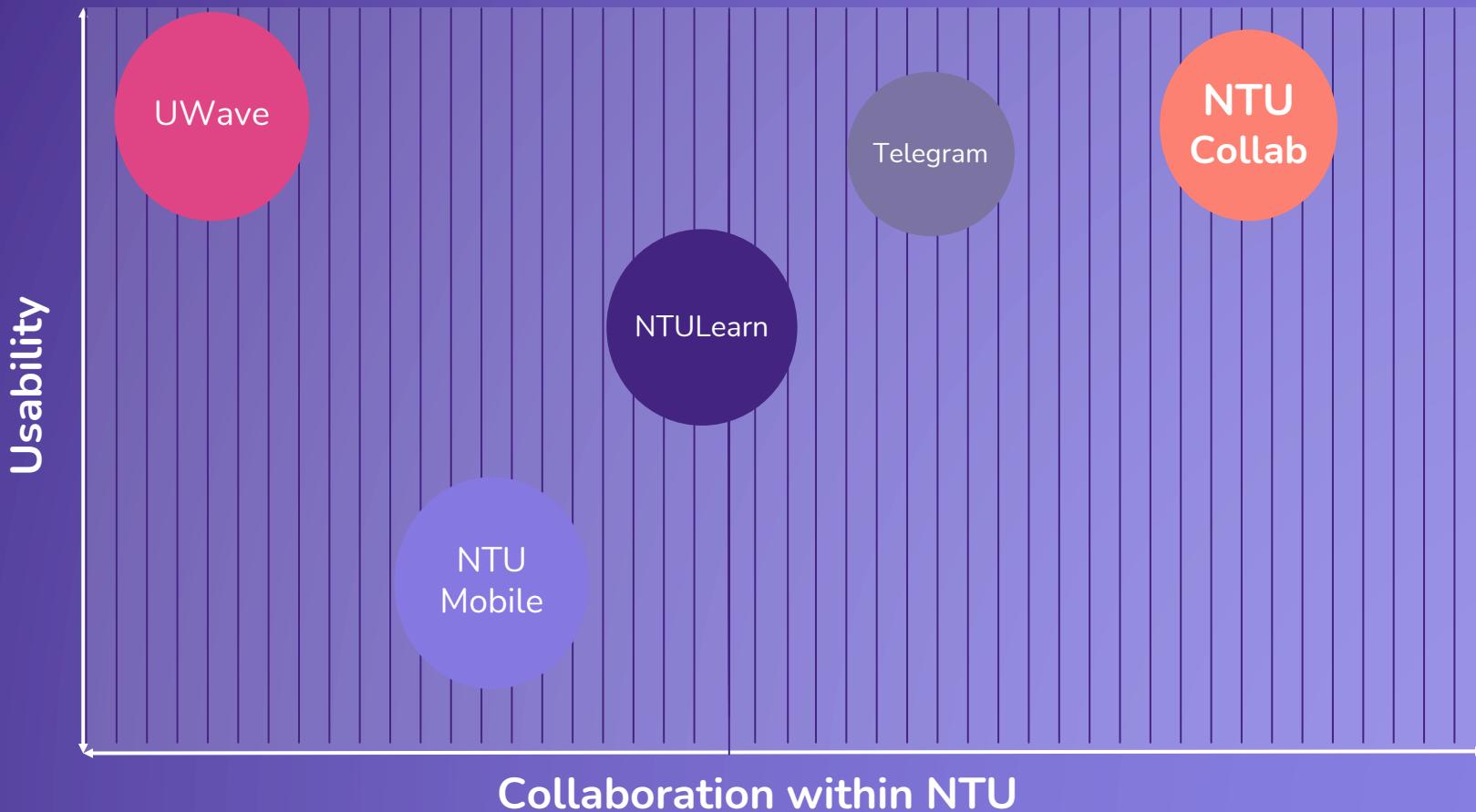




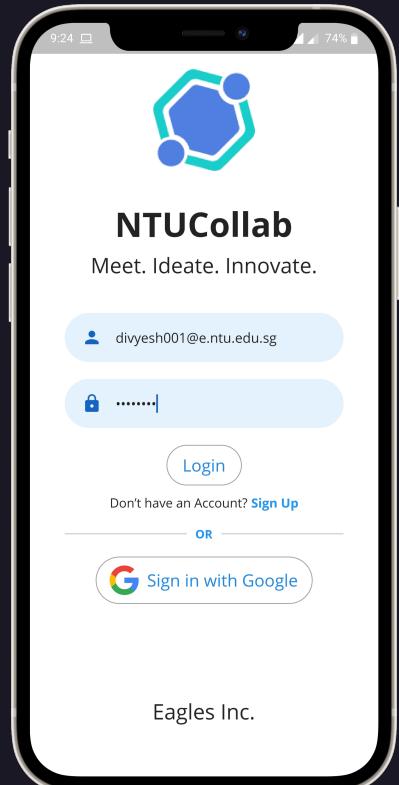
What is NTUCollab

- App to facilitate more interaction within NTU
- Recommends groups to join
- Modules, Clubs and Interest groups
- Allows user to create posts and rate for groups
- Create your own collaborative groups

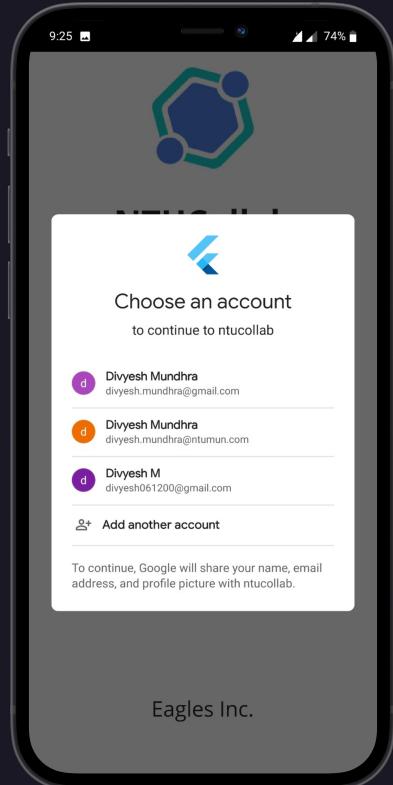
Competitor Matrix



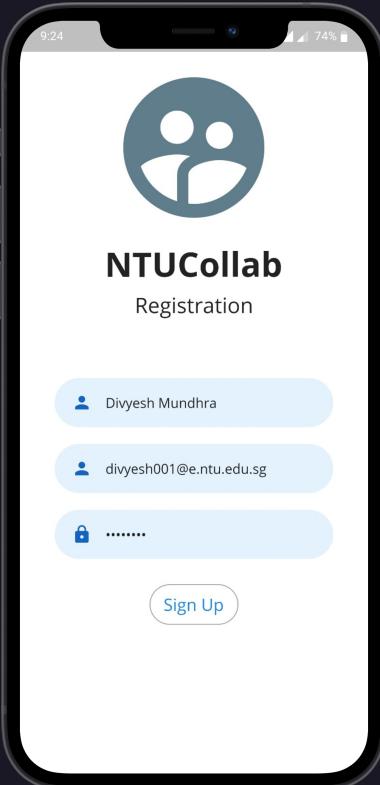
Login and Sign-Up Pages



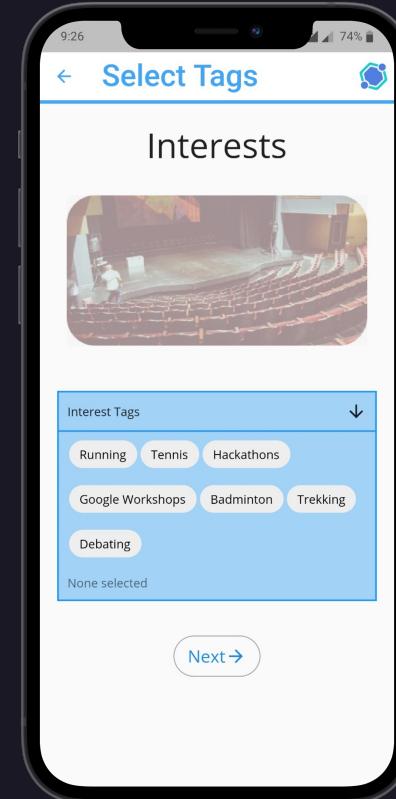
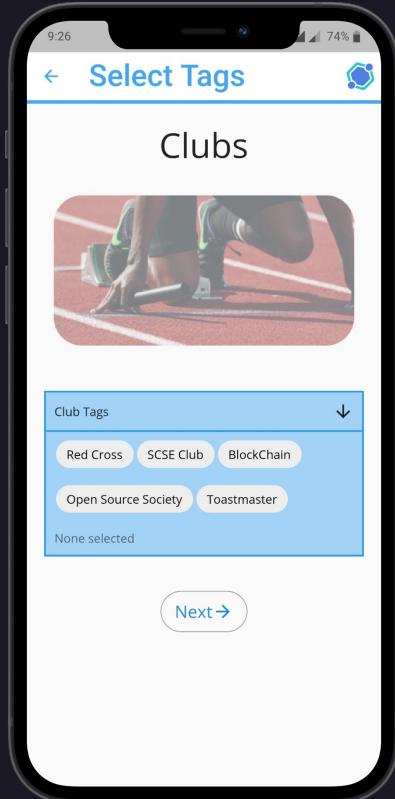
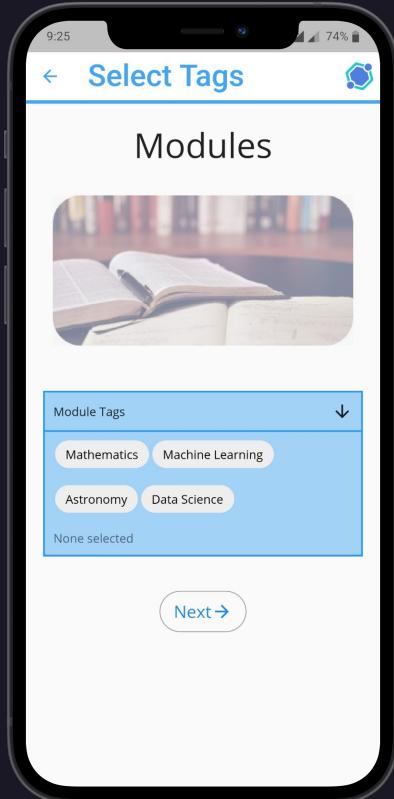
Eagles Inc.



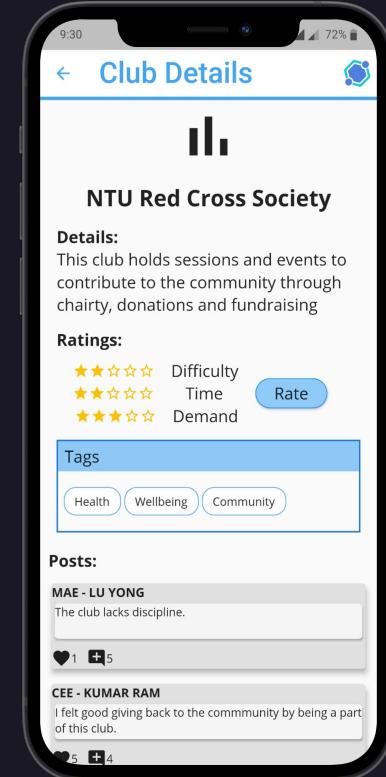
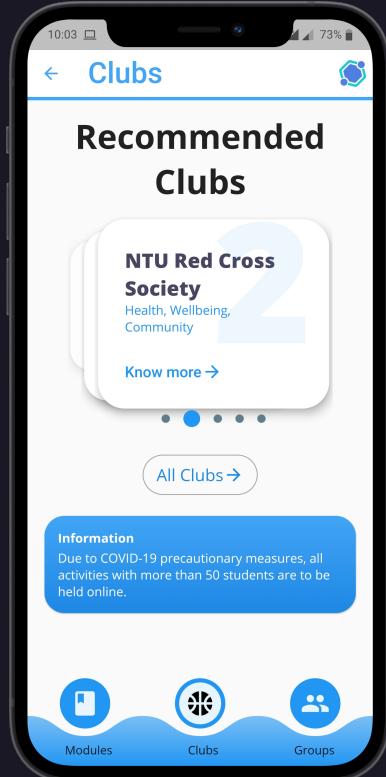
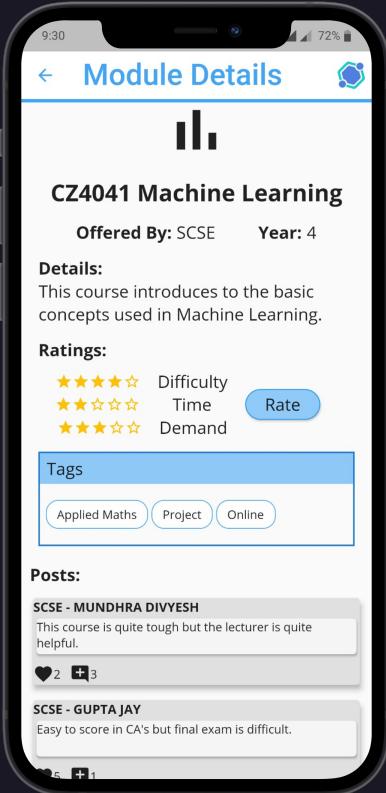
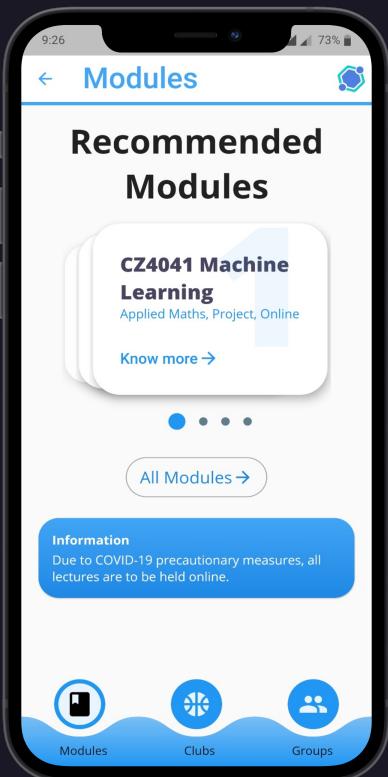
Eagles Inc.



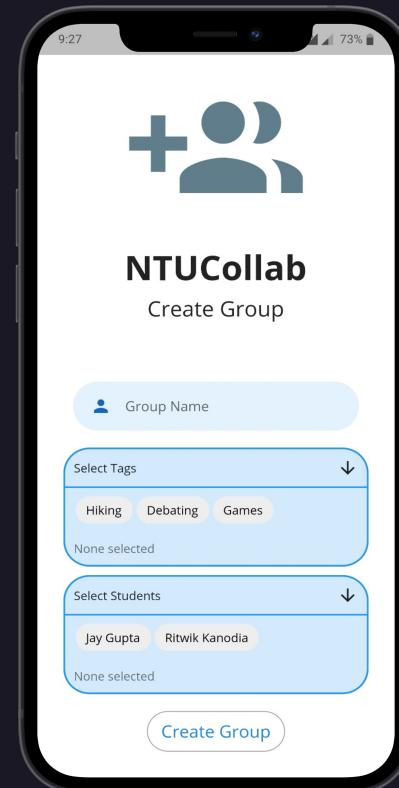
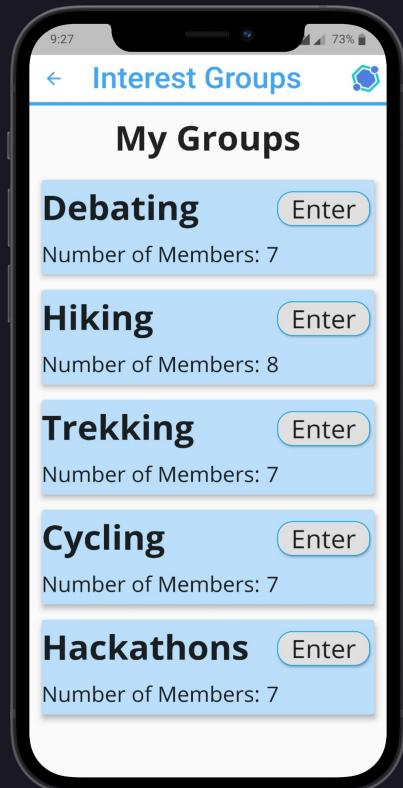
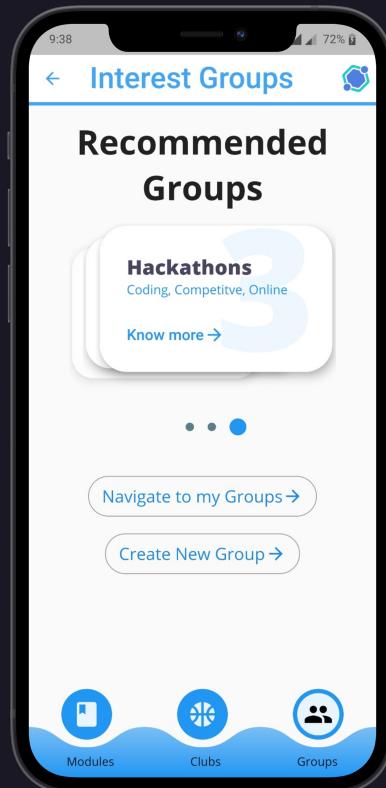
Selection of Tags



Modules and Clubs Section



Interest Groups Section

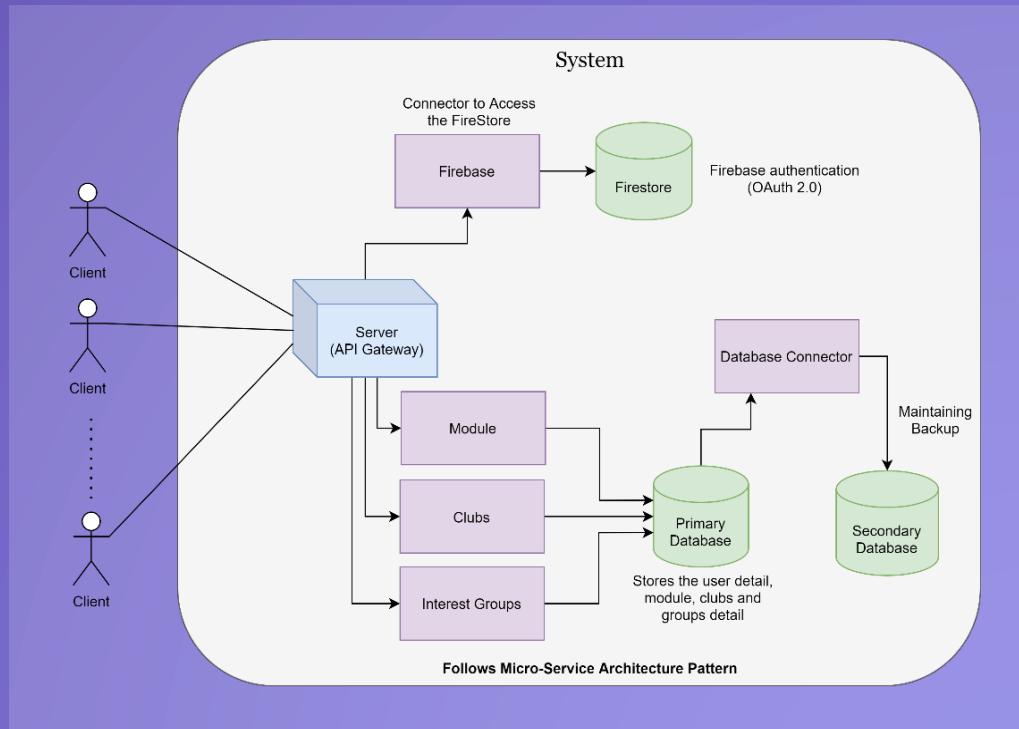


Design for Maintainability



System Design

- Microservice Architecture
- Firebase Firestore for Database
- Recommendation based on Tag Matching



Early Design

- Architecture considered for easy maintainability.
- New features constantly added hence needs to be adaptive.
- For usability, for the lack of actual target audience, we have used our internal team as testers.

Based on the design consideration, we have selected the Microservice Architecture.

System Design Consideration

Microservice Architectural Pattern

- Breaks the product into smaller component
- API Gateway to call different services
- Individual loosely coupled services

The architecture makes maintainability easier.
Any corrective and preventive maintenance can be
easily done without affecting the entire product.



Microservices



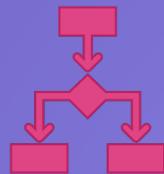
Database



- Google Firebase Firestore
- Allows the user to sign in with their Google Account using the Firebase authentication.



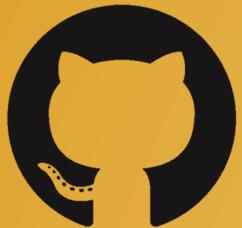
Recommendation Algorithm



- Uses tag matching
- Machine Learning based Algorithms to improve recommendation for future



Configuration Management



GitHub

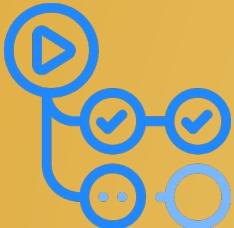


Microsoft
OneDrive



MEDIAWIKI

MediaWiki



GitHub
Actions

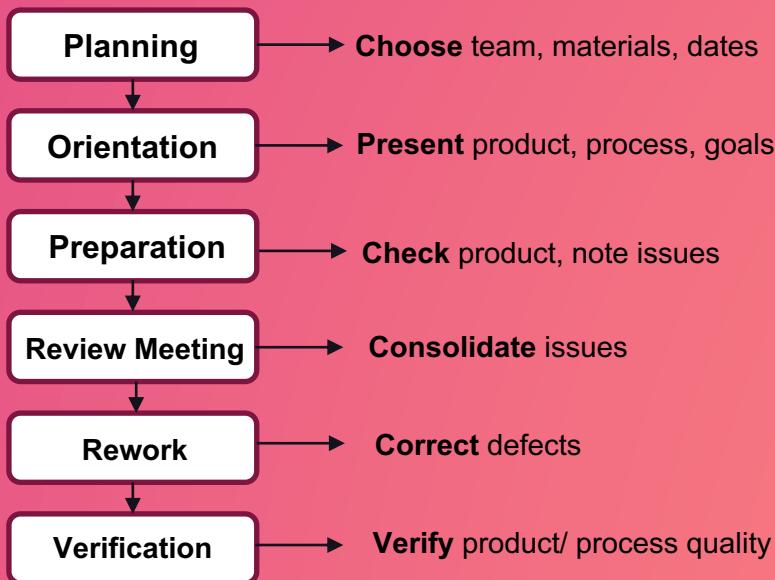


Microsoft
Teams

Software Quality Assurance



Technical Review



Formal Technical Review Process

Source: CZ3002 Lecture Notes – Dr Shen Zhiqi

Walkthroughs

- Informal walkthrough
- Developers not directly involved to perform review and raise issues found
- Iterative rectification of issues identified

Inspections

- Formal inspection from external reviewers with defined roles
- Identify issues and severity, for the entire project
- To be done repeatedly until all errors resolved

Management Review



- By Program Manager, Quality Manager and Lead Developer
- Every 2 weeks
- Include:
 - Status of schedules
 - Confirm requirements
 - Verify effectiveness
- Help meet time and budget constraints

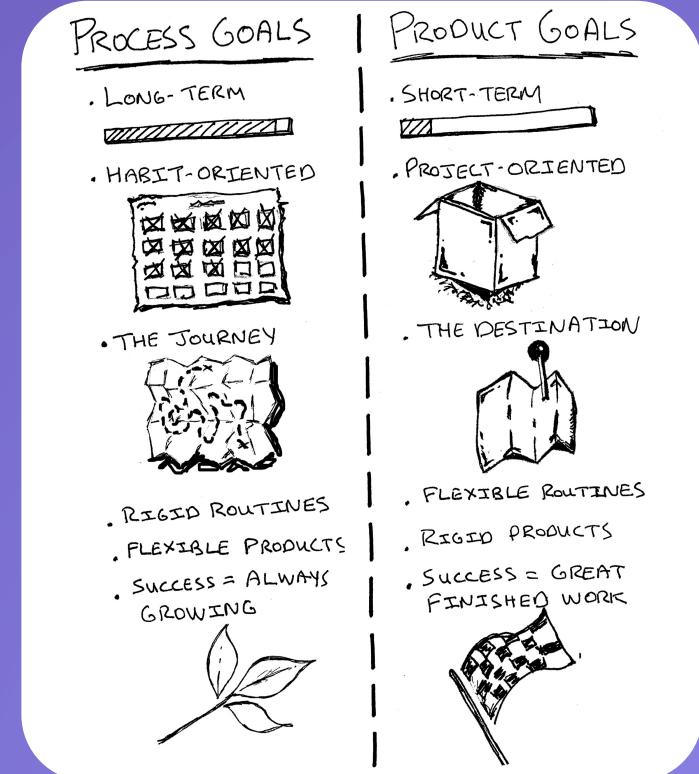
Audits



- 1 Consult people external to organization to conduct independent evaluation
- 2 Audit to be conducted less often than other reviews (approx. once per year)
- 3 Mainly conducted to ensure Team Eagles conforms to regulations, standard and plans such as the Project Plan



- Identify key areas of quality improvement
- Establish a **Quality Culture** in the team
- Traceability in product lifecycle, technical soundness, conformance to requirements etc. for NTUCollab



Commitment to Perform



- Briefings on project progress and organization policies
- QAM to gauge commitment from team and conformance to proper standards
- Increased Senior Management Sponsorship

Ability to perform



Resources

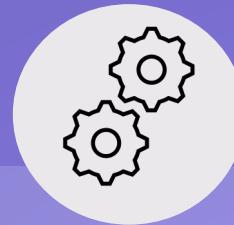
Access to all requisite Hardware & Software resources:

- PCs
- Cloud-infrastructure & services



Training

- Continued training as part of CZ3002
- The trainings are periodically revised according to the latest industry standards (E.g., IEEE)



Tools

- Project Management using Microsoft 365
- Version Control with Git
- VSCode Application Development Suite of tools

Activities Performed



Plans & procedures

- System Requirement Specification
- Quality Plan
- Project Plan
- Risk Management Plan
- Configuration Management Plan
- Report on Software Maintainability
- Change Management Plan
- Release Plan



Traceability

- All the documents tracked and stored using SVN
- Necessary corrective action through a change request workflow

Measurement, Analysis & Verification

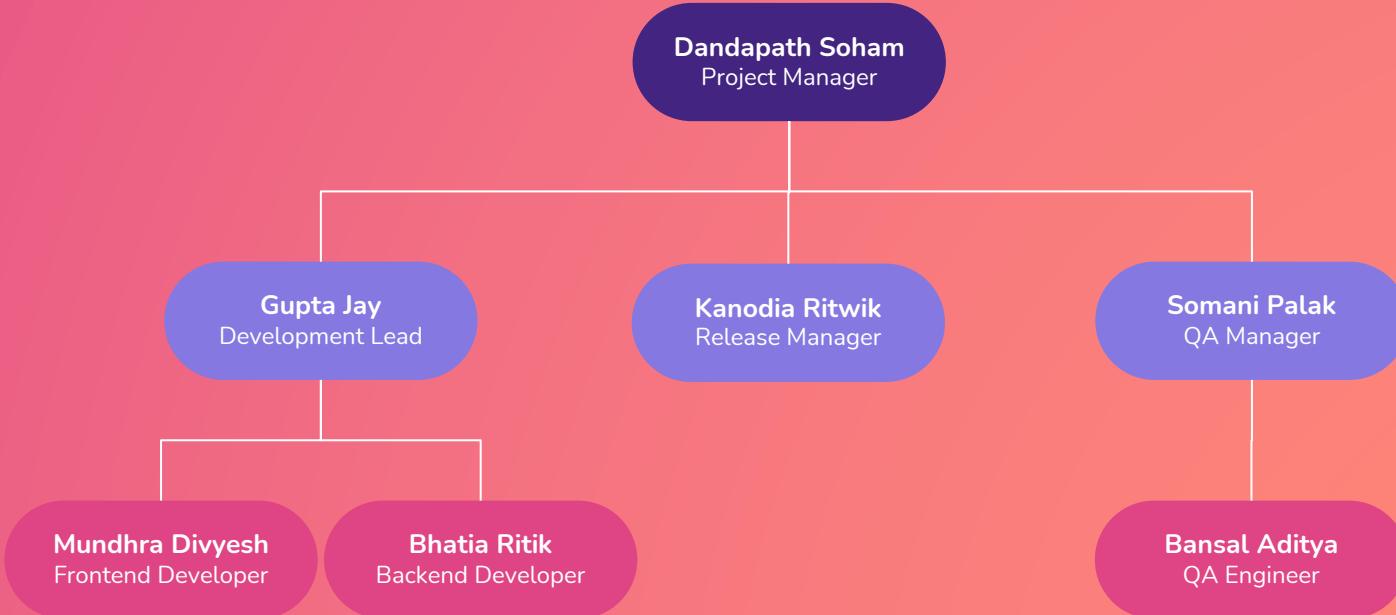


- Periodic assessment and reporting of the project's progress throughout the SDLC
- The performance metrics are made quantifiable and verifiable
- The results analysed by PM , and changes suggested based on the current state
- The amendments verified using audits by management and SQA

Project Management



Project Organization





Project Estimation



Function Points (1/2)

Characteristic	Complexity	Name
Inputs	Low	User Registration
		User Login
		Display of Groups
		User Discussion Forum
Logical Files	N/A	(Cloud Backend System)
Outputs	Low	Recommendation System
	Medium	Display of Groups
Inquiries	Medium	Search
		Filter



Function Points (2/2)

Characteristic	Complexity	Name
External Interfaces	Medium	User Registration User Login Display of Groups User Discussion Forum Edit Profile Create Group

Unadjusted Function Points

Characteristic	Low Complexity		Medium Complexity		High Complexity	
Inputs	9	x3	0	x4	0	x6
Outputs	1	x4	3	x5	0	x7
Inquiries	0	x3	6	x4	0	x6
Logical Files	0	x7	0	x10	0	x15
External Interfaces	0	x5	14	x7	0	x10
Unadjusted FP	31		137		0	
Total				168		



Adjusted Function Points

- Influence Factors – 8
- Total Influence Score – 19
- Influence Multiplier – $(19 \times 0.01) + 0.65 = 0.84$
- Adjusted FP – Unadjusted FP × Influence Multiplier
 $= (168 \times 0.84) = 141.12$



168

Unadjusted Function Points

141.12

Adjusted Function Points

6691 (47.41 LOC/FP)

LOC (Flutter – Dart)

Distribution of Effort (1/2)

1990's Industry data	Work Package	Distribution	Estimates (PH)
Preliminary Design (16%)	Project Plan	8%	34.534
	Requirement Specification	8%	34.534
Detailed Design (22%)	User Interface	5%	21.584
	Technical Architecture	11%	47.485
Code & Unit Testing (26%)	Data Modeling	6%	25.901
	Code & Unit Testing	20%	86.336
	Online Documentation	6%	25.901

Distribution of Effort (2/2)

1990's Industry data	Work Package	Distribution	Estimates (PH)
Integration & Tests (31%)	Integration and Quality Assurance	5%	21.584
Extrapolated total effort	-	-	410.096
3% Project Management	-	-	12.950
2% Contingency	-	-	8.636
Total Effort	-	-	431.68



Cost Estimates (1/3)

Hardware: Developer Workstations

1 – iPhone SE (to test NTUCollab on iOS) 1 – Google Pixel 4a (to test NTUCollab on Android)	
Single Core	\$649.00
1 GHz 4 GB RAM	\$500.00



Cost Estimates (2/3)

Software: GNU, Apache or another Free License software

Apache Web Server	\$0.00
Perl	\$0.00

Software License Provided by Third Party:

Microsoft Office 2000	\$0.00
Microsoft Project 2000	\$0.00
Products ESTIMATE Professional	\$0.00



Cost Estimates (3/3)

Others: Staff

8 Employees with 431.68 working hours with \$18.00/hour	\$7,720.24
--	------------

Others: Stationary

Paper, photocopying and other miscellaneous cost	\$40.00
---	---------



S\$8909.24

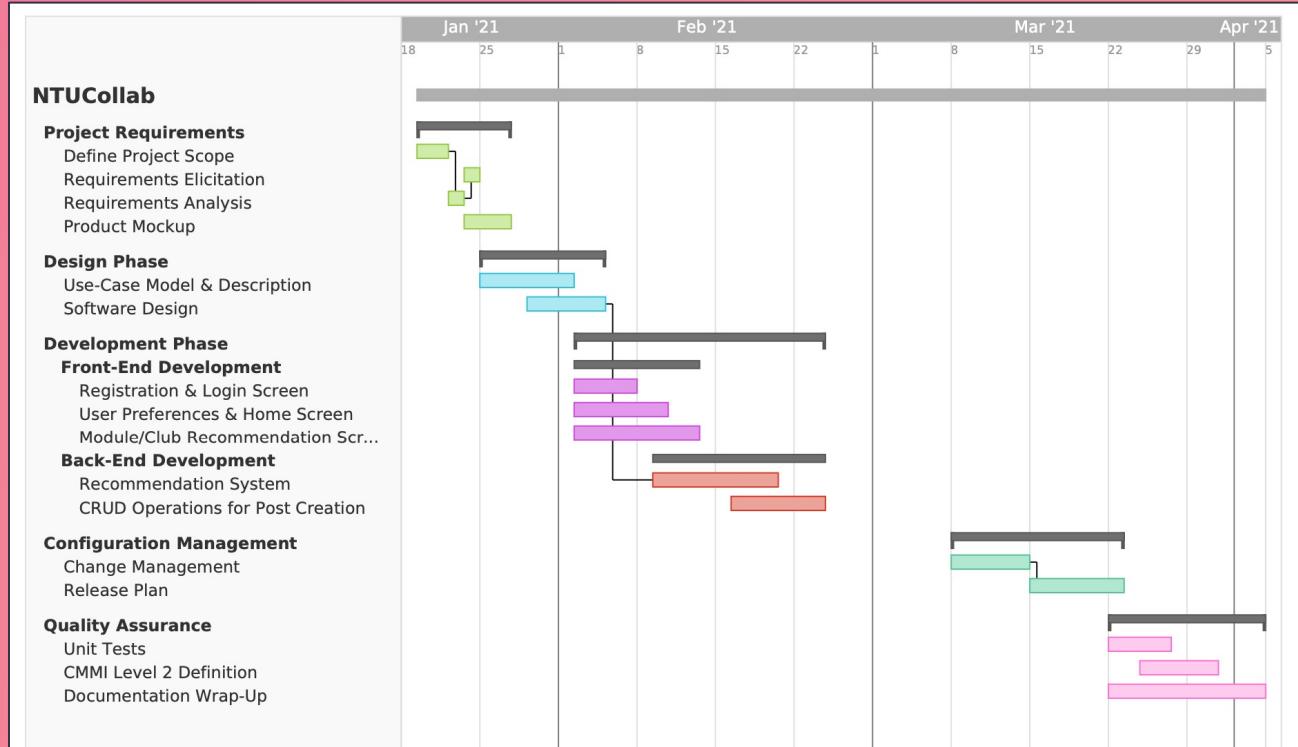
Total Cost



Activity Dependencies

Work Package No.	Description	Dependencies
X01	Project Planning	-
X02	Requirement Analysis and Documentation	-
X03	Requirement Analysis and Documentation	X02
X04	Prototype for User Interface	X01, X03
X05	System Design & Data Modelling	X04
X06	Individual Component Coding & Unit Testing	X05
X07	System Integration and Quality Assurance	X06
X08	Maintenance and Version Control	X07

Gantt chart



Risk Management



Types of Risk

1. Project Risks
2. Technical Risks
3. Business Risks
4. People Risks
5. Estimation Risks
6. Tools Risks

and many more ...



Risk Management Process



Risk Identification

Project team + Stakeholders

- 1 Technology
- 2 People
- 3 Organizational
- 4 Tools



Risk #1 - Technology

1. Google Login API malfunctions
2. Server crashes and essential data is loss
3. Reused software components might have bugs and defects which may break the code
4. The database used in the system cannot process as many transactions per second as expected



Risk #2 - People

1. Developers are occupied with multiple projects and may be unavailable to attend meetings
2. Conflicts between the developers and project manager
3. Team members falling ill
4. Undermined motivation
5. Inadequate domain knowledge



Risk #3 – Organizational



1. The project management might be restructured with different people responsible for the different parts
2. Course project might be revamped with different deliverables

Risk #4 – Tools

1. Code generated by mobile SDK might be inefficient for actual deployment
2. Use of Firebase for data storage might be insufficient for the user data
3. Deployment servers might not be scalable enough to meet the demands of peak user traffic



Risk Analysis

Identified risks need to be
assessed and prioritized

Qualitative

Quantitative



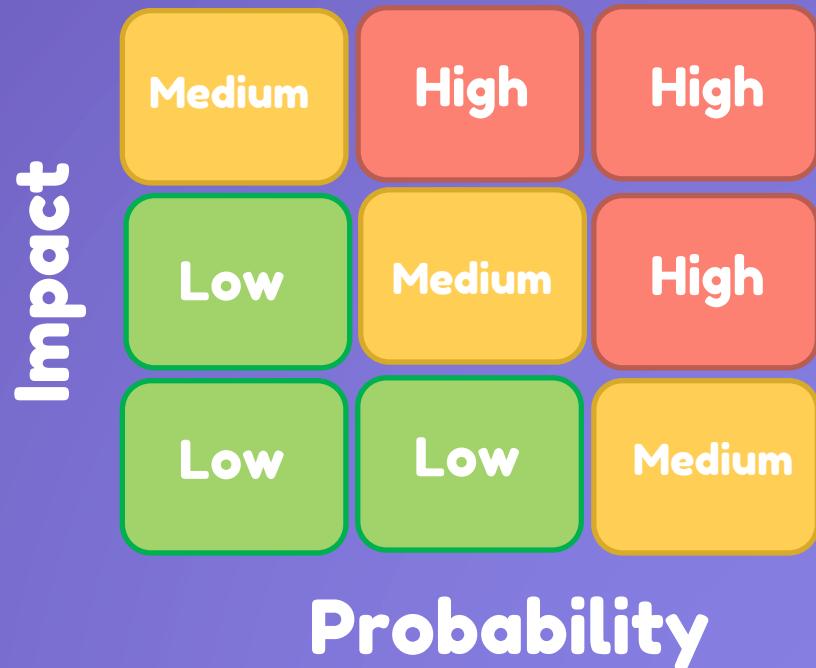
Qualitative Risk Analysis

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	1. Server Crash 2. Non-scalable deployment servers 3. Inefficient Code 4. Underestimation of Project Size	1. Reused Software Component Issues 2. High Rate of bugs & issues	
	Medium	1. Database Limits 2. API Malfunction 3. Developer Management Conflict 4. Undermined Motivation 5. Project Restructure 6. Inadequate Domain Knowledge 7. Planning Abandonment in high pressure 8. Under/Over Estimation of user base	1. Insufficient Firebase Storage 2. Requirements Changes 3. New use cases 4. Ill Staff 5. Time Underestimation 6. Multiple Commitments of Staff 7. Customer incorrect estimation of changes	
	Low	1. Scraper failure 2. Project Management Restructure	1. Course Structure Revamp	
		Low	Medium	High
			Probability	

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
Requirement Changes	3	3	1	7
Estimation	2	3	2	7

Risk Response Planning

1 Avoid

2 Mitigate

3 Accept

4 Transfer





Avoid

TECHNOLOGY

- Investment in high quality technology
- Highly redundant databases
- Highly scalable servers

Mitigate

PEOPLE

- More overlap between work of team members to ensure a substitute is available
- A team member to ensure team wellness, cohesion and monitoring of KPIs

TOOLS

- Thorough analysis of the tools to be used
- Remove defective components with reliable bought in tools

ESTIMATION

- Investigate use of a program to generate accurate estimation of timelines and project milestones

Accept

ORGANISATIONAL CHANGES

Organizational changes will have to be accepted and appropriate changes will have to be made accordingly

REQUIREMENT CHANGES

New mandatory requirement changes will be accepted, included in timeline and structured based on CRs



Risk Monitoring

Constantly identify, analyze, and plan for new risks

- Weekly Scrum Meetings
- Regular Test Runs



Monitoring previously identified risks

- Re-prioritize risks depending on probability and complexity

Thank You

Please contact our team admins
{ADITYA018, RITIK001}@e.ntu.edu.sg
for any questions or clarifications.

