Project Plan for NTUCollab
A Collaboratory platform for NTU Community

Dandapath Soham – Project Manager, Back-end Developer
Gupta Jay – Lead Developer, Release Engineer
Kanodia Ritwik – Front-end Developer, Release Engineer
Mundhra Divyesh – Front-end Lead Developer
Bhatia Ritik – Back-end Lead Developer
Somani Palak – QA Manager, QA Engineer
Bansal Aditya – QA Engineer

Team Eagles
School of Computer Science & Engineering
Nanyang Technological University, Singapore

Submitted to:
Dr Shen Zhiqi
School of Computer Science & Engineering
Nanyang Technological University, Singapore

Revision History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Bhatia Ritik	02/25/21	Soham Dandapath	02/24/21	Initial Project Estimates, Function Points, Effort, Cost, Duration and Schedule estimations
1.1	Kanodia Ritwik	02/26/21	Bhatia Ritik	02/27/21	Defined project description and scope
1.2	Dandapath Soham	02/27/21	Dandapath Soham	02/27/21	Added Gantt charts and detailed work packages
1.3	Kanodia Ritwik	02/28/21	Dandapath Soham	02/28/21	Added product checklist and best practices
1.4	Bansal Aditya	03/12/21	Bhatia Ritik	03/04/21	Added risk management, monitoring and control
1.5	Somani Palak	03/12/21	Dandapath Soham	03/04/21	Added quality assurance, monitoring and control
1.6	Bhatia Ritik	03/14/21	Bhatia Ritik	03/10/21	Final project plan, proof-read estimations, project organization, final formatting

Table of Contents

Contents

1. Introduction

1.1. Project Overview

NTUCollab is an application that provides a platform for NTU students to discuss and learn more about modules and clubs at NTU and meet their peers who share common interests. It captures the interests of the user upon registration and provides tailored recommendations for modules, clubs and interest groups. Users can either join the respective discussion groups of their interest or rate them.

NTUCollab will enrich the experience of NTU students, helping them make more informed decisions and pursue their interests.

1.2. Project Description and Scope

NTUCollab is a part of NTU suite of applications, such as U-Wave and NTU-Go, which aim at enhancing the campus lives of NTU students.

Collaborations with other products is a possibility which can be discussed in the near future.

While initially, NTU Collab will be limited to the students of NTU, it can later be expanded to other Singapore based universities as well.

The scope of NTUCollab aims to cover the academics, Co-curricular activities and general interests of NTU Students.

NTUCollab captures the interests of the user upon registration. These interests are from three above mentioned fields. Following this, it uses a recommendation algorithm to provide the user tailored recommendation in concurrent three fields namely, modules, clubs and interest groups.

However, for these recommendations to be apt, there are some constraints:

1. The user chooses at least three interest tags for all three fields.
2. These interest tags are indicative of the actual interest of the user.

The modules recommendation aims at helping students decide which modules would be a best fit for them. It does so by interfacing with STARS-planner by NTU and providing course details, pre-requisites and an average rating from other users of the application who may have done the course before. They also have the option of joining the pre-created discussion group of the respective module and posting their question there or taking part in discussions.

The interfacing with STARS is limited to extracting course descriptions and no other registration related data.

The club recommendations have similar objectives but for clubs at NTU and provide similar functionalities. The club description will be obtained from the clubs committee.

The interest groups recommendations aim at bringing together people with similar interests. We at NTUCollab believe that when the right people come together, big things can happen.

The section of the application will provide more freedom to the user as they can even create their own interest groups if it is not already present.

The users also have the option of searching and filtering specific groups in all three parts.

Since the application may consist sensitive user data, a secure registration system is implemented.

The messages are end to end encrypted to maintain user privacy.

As the application involves, further improvements and features maybe added based on user feedback, which is also obtained from the users, through the application from time to time.

2. Project Organization

2.1. Team Structure

This is structure of the team behind NTUCollab:

Roles	Members
Project Manager	Dandapath Soham
Lead Developer	Gupta Jay
Front-end Developer	Kanodia Ritwik, Mundhra Divyesh
Back-end Developer	Dandapath Soham, Bhatia Ritik
QA Manager	Somani Palak
QA Engineer	Bansal Aditya
Release Manager	Dandapath Soham
Release Engineer	Kanodia Ritwik

2.2. Roles and Responsibilities

Project Manager: Bansal Aditya

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

Lead Developer: Gupta Jay

- Leads the back end and front-end developers into the development of the project.
- Advises the project manager on all product development related decisions.
- Responsible for technical aspects of product release.

Front-end Developer: Kanodia Ritwik, Mundhra Divyesh

- Develop the front-end, user facing interfaces of NTUCollab based on project and customer requirements.
- Participate in the entire software development lifecycle, generating working product including relevant documentation, source code, unit and integration tests.

Back-end Developer: Ritik Bhatia, Dandapath Soham

- Develop the back end of NTUCollab, which includes the server, application and the database, based on the system requirements
- Participate in the entire software development lifecycle, generating working product including relevant documentation, source code, unit and integration tests.

Quality Assurance Manager: Somani Palak

- Ensures acceptable product and process quality
- Ensures proper implementation of Quality assurance processes

Quality Assurance Engineer: Bansal Aditya

- Designs testing strategies
- Creates and manages test plan
- Verify software requirements
- Executes test procedures

Release Manager: Dandapath Soham

- Manage the releases of Product.

Release Engineer: Kanodia Ritwik

- Create baselines and build and integrate changes for delivery.

2.3. Team Communication

NTUCollab communication channels include the following:

- Bi-weekly physical meetings are held on Tuesdays.
- Bi-weekly virtual meetings are held on other Tuesdays through Microsoft Teams.
- Group announcements and updates are sent through WhatsApp group.
- Split up into subgroups as necessary, in order to work more co-operatively on specific problems.

3. Process Definition

3.1. Lifecycle Model

The team intends to use the **Waterfall Model** throughout the NTUCollab project. This method is much more structured than other traditional methods as each phase must be completed before the next phase of development. It is relatively easy to accommodate any changes made to the software during the development process.

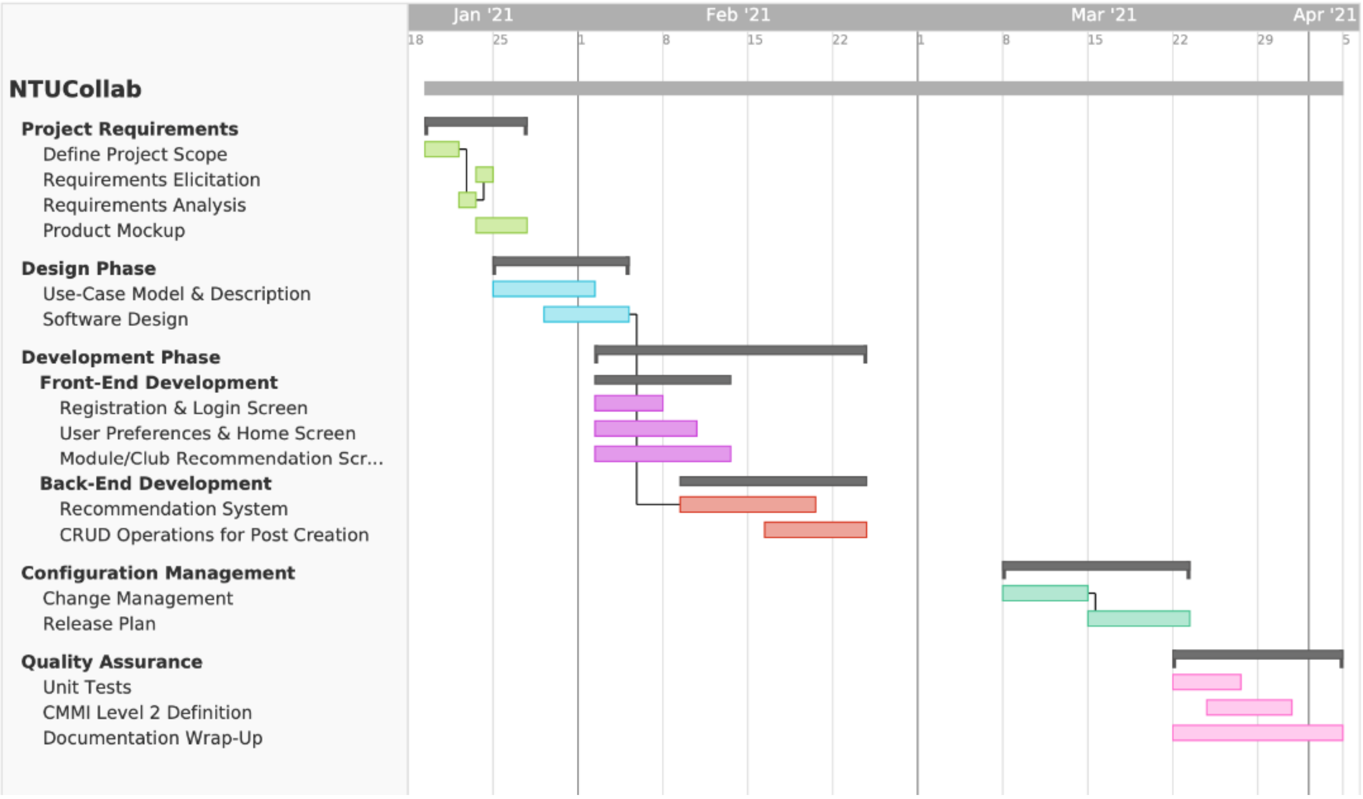
Given the financial constraints, using the Waterfall lifecycle model will help decrease costs, with the project being completely dependent on the project team with minimum client intervention.
Since this framework uses a step-by-step approach, errors are also easy to be identified and corrected. Early detection of errors, through appropriate Verification and Validation (quality assurance tests) before the completion of each stage, reduces the costs and time needed to correct them and improves the overall quality throughout the lifecycle.

We have chosen to avoid other methods such as Agile because they are not document driven and do not involve proper planning which are necessary for NTUCollab. Proper design, Quality management and development schedule is critical for NTUCollab.

We are also avoiding Spiral lifecycle models because they are complex and often drive up the costs. Time management is also difficult.

4. Schedule

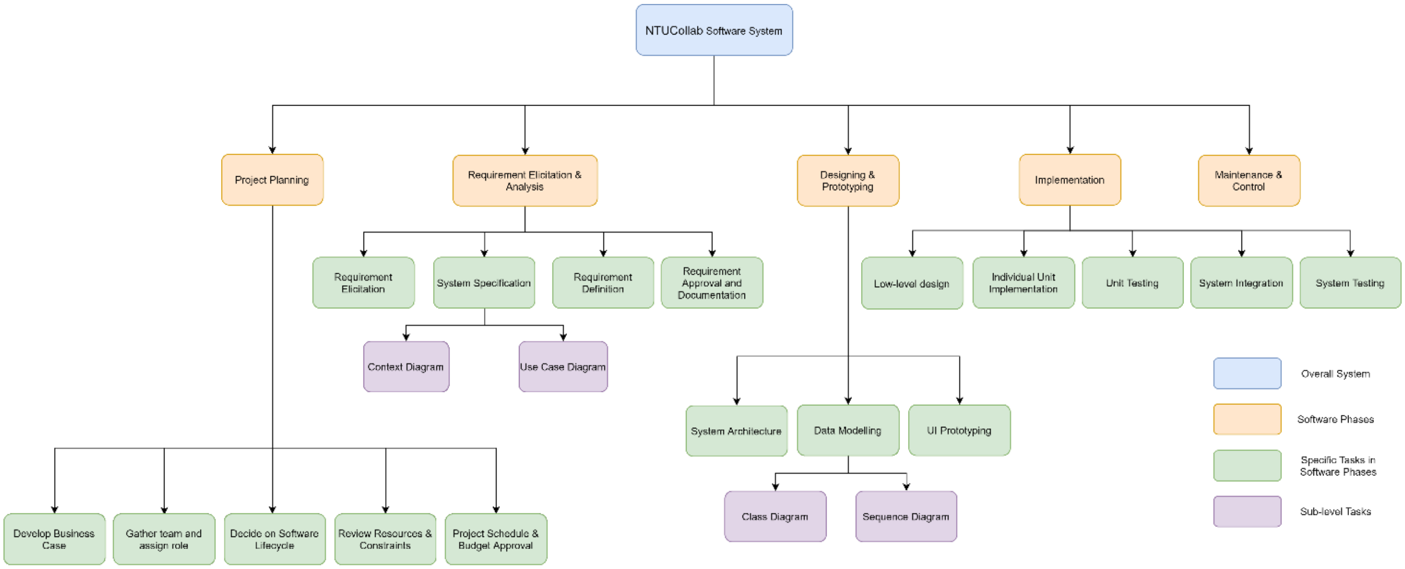
4.1. Activities, Dependencies and Schedule



4.2. Work Breakdown Structure

We are following a process-oriented work breakdown structure.

4.3. Work Packages



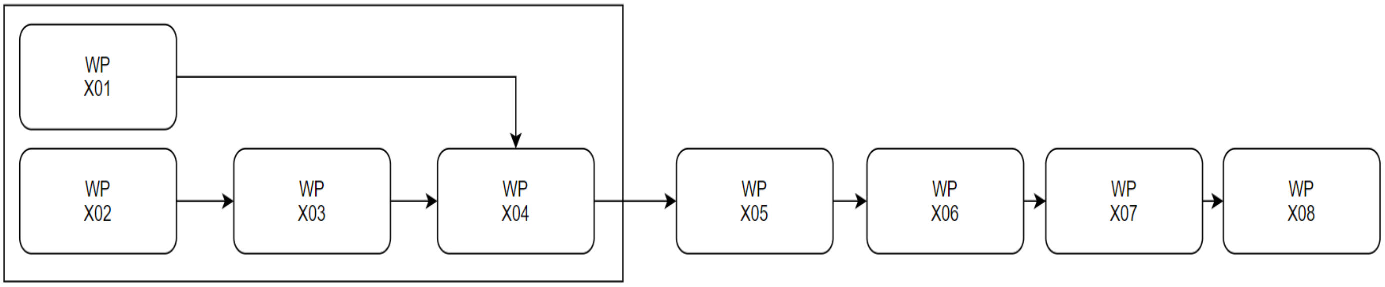
The project work is divided based on the different phases of the software development lifecycle.

1. Project Planning
2. Requirement Elicitation
3. Requirement Analysis and Documentation
4. Prototype for User Interface
5. System Design Architecture & Data Modelling
6. Individual Component Implementation and Unit Testing
7. System Integration and Quality Assurance
8. Maintenance and Version Control

4.4. Activity Dependencies

Work Package No.	Description	Duration	Dependencies
X01	Project Planning		--
X02	Requirement Elicitation		--
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

The activity network diagram is as follows:



4.5. Work Package Details

Project	NTUCollab Software System
Work Package	X01 – Project Planning (1 of 8)
Assigned To	**Bhatia Ritik**, Gupta Jay, Mundhra Divyesh, Somani Palak
Effort	35-person hours
Start Date	Jan 18th , 2021
Purpose	To determine the project overview which will be refined later in the later work packages
Inputs	None
Activities	This work package includes : 1. Development of the business case 2. Gathering team members and assigning individual roles 3. Providing a brief overview of the project 4. Its objectives and proposed project deliverables for individual software development phases 5. Developing a work breakdown structure 6. Project Schedule 7. Budget and Schedule approval The assigned team members will be transcribing the ideas proposed in the meetings into a formal documentation called the project plan.
Output	A written documentation containing the Project Plan Introduction.

Project	NTUCollab Software System
Work Package	X02 – Requirement Elicitation (2 of 8)
Assigned To	**Mundhra Divyesh**, Kanodia Ritwik, Bansal Aditya
Effort	17-person hours
Start Date	Jan 19th, 2021
Purpose	To gather requirements using the various requirement elicitation techniques

Inputs	Customer's requirements
Activities	This work package includes tasks: 1. Holding formal interviews with the customer to understand the requirements of the software 2. Brainstorming on the possible requirements that might be useful for the customers. 3. Building prototype to gather more requirements and build a common understanding on the requirements of the customers.
Output	A list of requirements gathered from interviews, brainstorming, prototyping and observations.

Project	NTUCollab Software System
Work Package	X03 – Requirement Analysis and Documentation (3 of 8)
Assigned To	**Mundhra Divyesh**, Kanodia Ritwik, Bansal Aditya
Effort	17-person hours
Start Date	Feb 1st, 2021
Purpose	To state a formal documentation based on the information gathered from the requirement elicitation.
Inputs	A list of requirements agreed from the previous (X02) work package.
Activities	The work package includes tasks: 1. Analysing the requirements gathered from the requirement elicitation and convert them into project deliverables.
Output	A formal documentation stating the requirements of the user and the estimated project deliverable agreed by both the customer and the development team.

Project	NTUCollab Software System
Work Package	X04 – Prototype for User Interface (4 of 8)

Assigned To	**Dandapath Soham,** Bhatia Ritik, Gupta Jay
Effort	22-person hours
Start Date	Feb 16th ,2021
Purpose	To build a prototype for user interface displaying the interaction between the system and the user and to mitigate and confusion or disagreement.
Inputs	User Information and Requirement document from work package X03
Activities	The work package includes : 1. Building a prototype to get user input, request for a function, display the dialog between the system and the user and finally to display the output of the request from the system. 2. Mitigate any disagreement between the customer and the developers.
Output	User Interface providing the front-end outlook of the system

Project	NTUCollab Software System
Work Package	X05 – System Design & Data Modelling (5 of 8)
Assigned To	**Gupta Jay**, Bhatia Ritk, Mundhra Divyesh, Kanodia Ritwik, Dandapath Soham
Effort	48-person hours
Start Date	March 23rd, 2021
Purpose	To do high level system architecture design and the flow for various components
Inputs	All the outputs from the previous work package (X01 – X04)

Activities	The work package includes tasks: 1. A high-level design of the entire system architecture. 2. Defining the various components required for the system 3. Modelling of the data in terms of communication between these various components 4. Deciding on the software system and hardware infrastructure required for the system such as the operating system built on, the database required for the storage. The designers must consider the various system design topics such as portability, maintainability for maintain the quality of the system adhering to quality assurance.
Output	High Level System Design Architecture Specification and Diagram

Project	NTUCollab Software System
Work Package	X06 – Individual Component Coding & Unit Testing (6 of 8)
Assigned To	**Somani Palak**, Bansal Aditya, Kanodia Ritwik, Gupta Jay
Effort	87-person hours
Start Date	Apr 6th, 2021
Purpose	To implement the system as per the UI design and the system architecture defined in the previous work packages. The package includes unit testing to test out the various quality assurance testing factors such as optimality and reliability for the individual components before the integration
Inputs	Outputs from the work package X05
Activities	The work package includes tasks: 1. Programmers implementing the modules according to the design specification defined in the system architecture.
Output	Source Code and header files

Project	NTUCollab Software System
Work Package	X07 – System Integration and Quality Assurance (7 of 8)
Assigned To	**Bansal Aditya**, Somani Palak, Bhatia Ritik, Dandapath Soham
Effort	134-person hours
Start Date	Apr 12th, 2021
Purpose	To integrate the various component into a single system and building the data flow between them. The package also contains fixing the syntactical and logical errors that may be come up during the integration. Black and white box testing would be conducted to check for logical errors. All the test plan report would be documented. Problems, if found would be fixed as soon as possible.
Inputs	Output from the work package X06
Activities	The work package includes tasks: 1. The integration team would simulate how a user may interact with the system. 2. Development of drivers, additional code during the integration testing as required for proper functioning of the system. 3. Testers will also examine and measure issues such as the performance, integrity and correctness.
Output	A Test Report.

Project	NTUCollab Software System
Work Package	X08 – Maintenance and Version Control (8 of 8)
Assigned To	**Kanodia Ritwik,** Dandapath Soham
Effort	12-person hours
Start Date	Apr 20th, 2021
Purpose	To maintain the system until the launch of the product to fix and issue or confusion from the customers.
Inputs	Output from work package X07

Activities	The work package includes: 1. Coders will unify and remove any discrepancy between the various versions of the software. 2. Observe the system for errors and remove the discrepancy if any raised by the customer. 3. Handover the customer the final software with proper documentation marking the end of the project.
Output	The final system software

5. Project Estimates

5.1. Code Size Estimation using Function Points

The code size is estimated using the Function Point method. This involved calculating the unadjusted function point based on the following parameters:

- Number of Inputs
- Number of Outputs
- Number of Inquiries
- Number of Logical Files
- Number of External Interfaces

After assigning a complexity to the tasks in each of the above 5 categories, we get the Unadjusted Function Point.

This is followed by taking several other drivers like Data Communications, Transaction Rate etc. into account to get the **influence multiplier** which when multiplied with the Unadjusted Function Point gives the **Adjusted Function Point**. The function point so obtained can be converted into lines of code based on research conducted over many years and with huge data.

Unadjusted Function Points

NTUCollab supports the following functions:

1. User Registration
2. User Login
3. Display of groups
4. Recommendation System
5. User Discussion Forum
6. Edit Profile
7. Create Group
8. Search
9. Filter

The measure of unadjusted function points is based on five primary component elements of these functions: Number of Inputs, Number of Outputs, Number of Inquiries, Number of Logical Files, and Number of External Interface. For each element, the complexity is either Low, Medium or High and accordingly, the weight of contribution to the unadjusted function point is assigned.

The detailed evaluation of the complexity is as follows:

User Registration

The steps included under User Registration are as follows:

- Details input by the user like Name, NTU Email ID, Course, Interest tags etc – **1 input**
- Details of the user are stored in cloud Firestore database (available on Google Firebase) – **1 external interface**

Hence, function points: **1 input** (low complexity)**, 1 external interface** (medium complexity)

User Login

The steps included under User Login are as follows:

- Input of details by the user for validation – **1 input**
- Verification of input details against details stored in cloud Firestore database – **1 external interface**

Hence, function points: **1 input** (low complexity)**, 1 external interface** (medium complexity)

Display of groups

The steps included under Display of Groups are as follows:

- Display of recommended groups under 3 categories: Modules, Groups and Clubs
- Display and edit rating under 3 categories: Time Commitment, Difficulty and Demand. Store the saved ratings in cloud Firestore database – **3 inputs, 3 external interfaces**
- Display the information for each in the respective groups after pulling information from cloud Firestore database – **3 outputs, 3 external interfaces**
- Module information to be pulled from an external interface (STARS) – **1 external interface**

Hence, function points: **3 inputs** (low complexity)**, 3 outputs** (medium complexity)**, 7 external interfaces** (medium complexity)

Recommendation System

The steps included under Recommendation System are as follows:

- Processing of interest tags for a user and suggesting appropriate groups – **1 output**
- Gives user the option to join or reject the recommendation of the system

Hence, function points: **1 output** (low complexity)

User Discussion Forum

The steps included under User Discussion Forum are as follows:

- Post/chat in discussion forum of each type of group (Module/Club/Group) and store the same in the cloud database – **3 inputs, 3 external interfaces**
- The discussion forum shall provide support for text and media input

Hence, function points: **3 inputs** (low complexity)**, 3 external interfaces** (medium complexity)

Edit Profile

The steps included under Edit Profile are as follows:

- Allow the user to change profile information like name, interest tags, photo etc. – **1 input**
- Store the changes in the cloud Firestore database – **1 external interface**

Hence, function points: **1 input** (low complexity)**, 1 external interface** (medium complexity)

Create Group

The steps included under Create Group are as follows:

- Details of the group like name etc. input by the user – **1 input**
- Store the new group in the cloud Firestore database – **1 external inteface**

Hence, function points: **1 input** (low complexity)**, 1 external interface** (medium complexity)

Search

The steps included under Search are as follows:

- Input by the user to search the respective Modules/Clubs/Groups – **3 inquiries**

Hence, function points: **3 inquiries** (medium complexity)

Filter

The steps included under Filter are as follows:

- Allow user to filter results based certain criteria, in each of Modules/Clubs/Groups – **3 inquiries**

Hence, function points: **3 inquiries** (medium complexity)

Based on the complexity (low / medium / high) of each function point, the unadjusted function point can be calculated as follows:

Characteristic	Low Priority		Medium Priority		High Priority	
Inputs	9	× 3	0	× 4	0	× 6
Outputs	1	× 4	3	× 5	0	× 7
Inquiries	0	× 3	6	× 4	0	× 6
Logical Files	0	× 7	0	× 10	0	× 15
Interfaces	0	× 5	14	× 7	0	× 10
Unadjusted FP	31		137		0	
Total = Low + Medium + High	168					

Adjusted Function Points

Influence Factor	Score	Details
Data Communications	4	Application involves significant data communication between user and cloud Firestore database
Distributed Functions	0	The mobile application does not involve distributed processing
Performance	2	Response time or throughput is moderately important for the application to prevent long waiting times for the user
Heavily used	1	The application is expected to have little load
Transaction rate	0	Peak transaction rates are not expected

On-line data entry	3	All the data transactions are online and need to be updated continuously on the cloud database
End-user efficiency	0	End user efficiency is not a major consideration
On-line data update	0	Online update of internal files is not included
Complex processing	3	Complex processing at certain steps like searching/filtering and calculating the ratings are important parts of the app
Reusability	3	Many parts of the application are expected to be reusable as logic for Modules/Clubs/Groups are extremely similar
Installation Ease	0	No special considerations for ease of installation have been made
Operational Ease	1	No special consideration given
Multiple sites	0	Requirements do not include considering availability on multiple sites
Facilitate change	2	NTUCollab should be designed to facilitate change as it an application in the initial phase and is expected to have major changes in the future
Total score	19	
Influence Multiplier = Total score × 0.01 + 0.65 = 19 × 0.01 + 0.65 = 0.84		
Adjusted FP = Unadjusted FP × Influence Multiplier = 168 × 0.84 = 141.12		

The legend used for scoring is as follows:

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

Lines of Code

According to Capers Jones statistics, each Function Point requires 47.41 lines of code if the application is implemented using **Dart** (the Flutter framework used for the development of NTUCollab uses Dart programming language).
Therefore, we have: **Lines of Code** = 141.12 FP × 47.41 LOC/FP = **6691 LOC**

5.2. Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for NTUCollab, we use function points as the basis to calculate Effort, Duration and Team size. Various assumptions have been made to expand the estimates, include generality and get the best estimate for each of the parameters to be calculated.

- All the group members are expected to contribute on all 7 days of the week
- Effort = Size/Production Rate = (6691 LOC) / (62 LOC/PD) = 107.92 PD (Used: **CND 97**)
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (107.92) $^{1/3}$ = 14.28 Days
- Initial schedule = 14.28 Days / 7 days a week = 2.04 Weeks
- Team size = 107.92 PD / 14.28 D = 7.557 P = 8 Persons
- Working hours include 4 hours in a working day.
- Total person-hours (PH) = 107.92 PD × 4 hours = 431.68 PH

Distribution of Effort

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
	Data Modeling	6%	25.901
Code & Unit Testing - **26 %**	Code & Unit testing	20%	86.336
	Online Documentation	6%	25.901
Integration & Test – **31%**	Integration & Quality Assurance	31%	133.821
	Extrapolated total effort		410.096
	3% for project management		12.950
	2% for contingency		8.636
	Total effort		431.68

The above duration estimates assume that each team member is committed fully and works equally

5.3. Cost Estimates

Hardware: Developer workstations

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

Software: GNU, Apache or other 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

Others:

Staff:

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

Stationary:

Paper, photocopying and other miscellaneous cost	$40.00

Total: $8909.24

The customer is expected to bear the cost for the hardware (a stable network and client computers) to run NTUCollab. The NTUCollab team is in no way responsible for providing the hardware for to run the application. All the hardware requirements and the costs listed above are solely for the purpose of development of the application as required.

Further, the two phones (iPhone and Google Pixel 4a) with the specifications mentioned above will also not be provided by the NTUCollab team as they are meant for testing and debugging purposes by the team. To use the application, the user is expected to install and download it from the relevant application store for their device (AppStore / PlayStore).

6. Product Checklist

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

Project Deliverable	Estimated Deadline
Project Proposal	Feb 2nd ,2021
Use case model	Feb 2nd ,2021
System Requirement Specifications	Feb 16th ,2021
Quality Plan	Feb 16th ,2021
Project Plan	Feb 16th ,2021
Risk Management Plan	March 23rd, 2021
Prototype Development	March 23rd, 2021
Prototype Demo Video and Slides	Mar 23th,2021
Design Report on Software Maintainability	Apr 6th, 2021
Configuration Management Plan	Apr 6th, 2021
Change Management Plan	Apr 6th, 2021
Release Plan	Apr 6th, 2021
Module/System Test Plan	Apr 20th , 2021
Test Cases and Requirements Test Coverage report	Apr 20th , 2021
CMMI level 2 definition	Apr 20th 2021
System Release (Demo)	Apr 25th, 2001

7. Best Practice Checklist

Practice
All actions and strategies, whether it be related to the product or the process should be documented. The documents should be formatted well and easy to follow.
Pay attention to requirements, check for ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification.
The complexity should be managed well. Try to keep it simple by following these guidelines: Minimize interfaces between modules, procedures and data. Minimize interfaces between people, otherwise exponential communication cost Avoid fancy product functions, design as long as the functionality meets the customer requirements
Require Visibility. We must see what we build otherwise we can measure the progress and take management action. This includes: the manager must have good communication with his or her employees; require developers to make code available for review; review design for appropriateness.
Plan for continuous change. We must: • All manuals designs, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes • New revisions should be approved before being made and checked for quality and compliance after being made • Use a configuration management system and make processes • Required maintenance
Don't under estimate. We must be careful to obtain accurate estimates for: time, effort, overhead, meeting time, and especially effort on integration, testing, documentation and maintenance.
Code reviews are a much more efficient method to find software defects. Plan and manage code reviews between team members
Software testing will use both black box and white box testing. It will involve unit, functional, integrating and acceptance testing.

8. Risk Management

Besides the general risk management, the following risks have been identified for the NTUCollab:

8.1. Database Limits

- *Impact Severity:* Medium
- *Probability:* 25%
- *Impacts:* Depending on the peak user traffic and the number of requests, the application might serve to respond to all the users, leading to delayed or no responses.
- *Risk Reduction:* investment in highly redundant, reliable and elastic database infrastructure

8.2. Google login API Malfunction

- *Impact Severity:* Low
- *Probability:* 15%
- *Impacts:* The users will not be able to login using their google credentials
- *Risk Reduction:* An alternative and backup login option will be provided for users to login using their email addresses to create an account with our application.

8.3. STARS Planner Scraper Failure

- *Impact Severity:* Medium
- *Probability:* 25%
- *Impacts:* The application will not be able fetch the modules data, to provide recommendations. (Only the module recommendation component will fail, other recommendations will continue to work).
- *Risk Reduction:* Every 2 weeks the course module data, will be backed up on the application side, to provide some recommendations, in case of such a failure.

8.4. Undermined motivation among developers and managers

- *Impact Severity:* Medium
- *Probability:* 10%
- *Impacts:* The project might not be completed on time with required quality
- *Risk Reduction:* In the SCRUM framework of project management being used in the project, a SCRUM Master will be assigned to ensure the wellbeing of developers and smooth communication between the manager and developers.

8.5. More changes to requirements than anticipated

- *Impact Severity:* High
- *Probability:* 25%
- *Impacts:* Depending on the stage at which changes occur, could range from needing to update the requirements documentation to a needing to do a complete redesign.
- *Risk Reduction:* Be rigorous in eliciting requirements. Make customer aware of potential repercussions of requirement changes.

8.6. Specification Delays

- *Impact Severity*: High
- *Probability:* 15%
- *Impacts:* Delay in finalizing the specification will push the schedule for all following stages of the project.
- *Risk Reduction*: Monitor progress of specification carefully.

8.7. System size underestimated

- *Impact Severity:* Moderate
- *Probability:* 30%
- *Impacts:* More work will need to be spent on design and coding; could negatively impact schedule.
- *Risk Reduction:* Update estimates often as project progresses.

8.8. Staff leaving before project complete

- *Impact Severity:* Extreme
- *Probability:* 5%
- *Impacts:* There would be more work for remaining employees, and any specialized skills or knowledge would be lost.
- *Risk Reduction:* Offer benefits and incentives to staff.

8.9. Problems co-ordinating 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 in section 2.3

8.10. Customer cancels project

- *Impact Severity:* Super extreme
- *Probability:* 1%
- *Impacts:* All work will have been wasted.
- *Risk Reduction:* Keep in close contact with customer. Ensure that they have some market research indicating demand for the product

9. Quality Assurance

The project will achieve the quality assurance by following the standard set by the company. The specific procedures and details shall be provided in the Quality Plan.

Specific test procedures and details shall be provided in the Module/System Test Plan.

In addition, the NTUCollab shall make use of following testing methodologies:

- **Unit Testing**
  All code will be unit tested to ensure that the individual unit (class) performs the required functions and outputs the proper results and data. Proper results are determined by using the design limits of the calling (client) function as specified in the design specification defining the called (server) function. Unit testing is typically white box testing and may require the use of software stubs and symbolic debuggers. This testing helps ensure proper operation of a module because tests are generated with knowledge of the internal workings of the module. To summarize, individual classes will be tested to ensure reliability and functionality within a unit level. Furthermore, testing module will be created before the tested code to ensure that the code is testable.

- **Integration Testing**
  There are two levels of integration testing. One level is the process of testing a software capability e.g., being able to send a message via a DMA port. or the ability to acquire a row of CCD data. During this level, each module is treated as a black box, while conflicts between functions or classes and between software and appropriate hardware are resolved. A second level of integration testing occurs when sufficient modules have been integrated to demonstrate a scenario e.g., type of science mode or the ability to queue and receive commands. During this phase, composite builds, or baselines, of the software are married to the engineering versions of the hardware to evaluate the combined hardware/software performance for each operational function. Both hardware and software documentation are reworked as necessary. To summarize, several classes will be tested together to ensure sufficient execution and compliance with the requirements after integration. Integration testing will be performed before beta release.

- **System Testing**
  The purpose of system (stress) testing is to identify the operational envelope of the instrument. The whole system shall be used for system testing to ensure all requirements is satisfied, and reliability will be included in the testing to measure successful rate of message delivery. System testing will be performed before beta release.

- **In-Place Testing**
  Involves testing of the whole system as a unit.

Furthermore, the following standard metrics will be collected, reported, and maintained in software quality assurance:

- **Defect Density**

  It measures the number of defects relative to the software size expressed as lines of code or function point. Simply put, it measures the code quality per unit. The SQ personnel are responsible for collecting and tracking the number of defects detected per 250 lines of code in NTUCollab and suggest amends to minimize the same.

- **Length of Code**

  The length of code is a very simple yet common metric. It is based on the notion that longer the code, the more difficult it is to understand it. The length of code can be minimized by improving reusability of the code through object-oriented practices like using functions to modularize the code, inheritance, interfaces, abstract classes etc. The SQ personnel must ensure that they frequently monitor the length of the code of NTUCollab software, detect if there is unnecessary code that can be removed and suggest ways to further keep the length of the code to a minimum. The standard unit used for the measurement of the length of code for NTUCollab is KSLOC (Thousand Source Lines of Code)

- **Cyclomatic Complexity**

  It is an important software metric and is used to indicate the complexity of a program, by looking at the source code. It is a quantitative measure of the number of linearly independent paths in the program’s source code. SQ personnel must monitor NTUCollab’s source code and ensure the cyclomatic complexity is low, that is, not many parts of the program are interdependent on each other. If this complexity is high, the Project Management must be notified of the same and asked to modularize the code into independent functional sections.

- **Percent delinquent fixes**

  It is a software maintainability metric and is simply the percentage of all fixes in a time interval that are defective. It is imperative to keep this at a minimum through thorough testing of a fix. The SQ personnel are responsible for tracking the fixes in NTUCollab and relay any defective fixes to the Project Management team, as a high number of defective fixes points to a larger issue in the codebase, highlighting to many other potential defects.

- **Mean Time to Failure (MTTF)**

  It is the average time between failures. It provides an idea of how frequent errors might occur in the software. With regards to the project NTUCollab, the SQ personnel will track the MTTF in various scenarios of the application like excessive load, parallel access of resources used by the application etc.

- **Problems per User-Month in UAT**

  User Acceptance Testing (UAT) is an important metric since it is used to simulate the user experience of using the application. If the number of problems detected during UAT exceeds a particular threshold, then it is a cause of worry and it is important to immediately resolve such issues as users might not adopt the application otherwise. During the UAT phase of NTUCollab, the SQ personnel must take note of all errors encountered and track the average number of errors per month. If this number is high, they should immediately notify the Project Management team, along with a detailed log of the issues faced, for effective troubleshooting.

Lastly, the following are the most important qualities required in the NTUCollab software:

- **Maintainability**
  The capability of the software product to be modified. Modifications may include corrections, improvements, or adaptions of the software to changes in the requirements. The coding practices used for developing NTUCollab should be such that they ensure the code is maintainable. Developers should maintain maintenance logs to refer to similar scenarios in the future. The software should be designed such that it is easy to modify and change as per changing requirements. Further, errors and bugs should be easy and use minimal steps to troubleshoot and resolve.

- **Usability**
  The capability of the software product to be understood, learnt, used and attractive to the user when used under specified conditions. It is imperative that NTUCollab be usable by having a simple user interface, so that the user is not overwhelmed. The users of our mobile application will accept it only if they find it easy to navigate and use.

- **Reliability**
  The capability of the software product to maintain its level of performance under stated conditions for a stated period. NTUCollab should be a reliable software. Its code design should not include tight coupling or singletons and should make use of more reusable code. This will help to improve its performance and minimize downtime.

- **Functionality**
  The capability of the software product to provide functions which meet stated and implied needs when the software is used under specified conditions. It is important that NTUCollab is a fully functional mobile application and works as expected by the user. It should satisfy all functional requirements and should be traceable if any error arises. Further, it is compulsory that NTUCollab has cross-platform support, that is, available to users of both Android and iOS, to cater to the needs of a wider audience.

10. Monitoring and Control

Many procedures are required to be able to successfully monitor the progress of a software project.

Some of the most important procedures are:

- **Identification of major project risks**
  Early identification of major risks to the project allows for placement of preventative measures before problems can develop. Major risks have been identified in the Risk Management section of this document, along with the measures being taken to avoid them.

- **Regular reviews of project progress**
  Throughout the duration of the NTUCollab project, the team shall meet weekly to review the progress of all project tasks, including management, planning, analysis, development, and testing.

- **Timeline Planning and task decomposition**
  This document outlines an estimated timeline for the project. A reasonably accurate timeline can be assembled by hierarchically decomposing tasks into measurable subcomponents and estimating requirements for each. At the same time, this decomposition can assist in task assignment and balancing. Throughout the implementation phase, these subcomponents can allow for fine-grained measurement of progress. Project subcomponents and timeline estimates are included in the Estimates and Work Breakdown Structure sections of this document.

- **Risk Management Process**
  SQ personnel are accountable to ensure that all risk management measures are put in place and adopted, to mitigate risks and minimize the impact, if any. This includes drafting and relaying a formal risk management plan and enforcing the adoption of the same. All edge cases and potential risks should be flagged, and the team should prepare for such contingency scenarios in advance. Possible risk scenarios include running behind schedule, lack of resources for implementation of some features, lack of developer experience regarding some part of the application etc.

- **Risk Identification and analysis**
  Risk will be actively identified, considering potential risks to and from project team, stake holders along with extrinsic factors, like environmental, cultural, and political factors capable of adversely impacting the project. The Project proposal, deliverables , constraints, schedule, and budget will be scrutinized to mitigate risks. Thereafter risks will be analysed in two broad categories of quantitative and qualitative risks. The likelihood and the probability if each identified risk will be calculated, into categories of low, medium, and high. Further, quantitative metrics to provide an objective numerical rating to the risk will be estimated. All these will be clearly documented in the risk management plan.

- **Risk Response Planning, Monitoring & Control**
  The analysed risk will be categorized into varying level of criticality of red, yellow, and green zones. The Red and yellow risks will be assigned to individual personnel, to maintain and ensure each risk is carefully mitigated. Through framework of avoid, mitigate, accept and transfer will be used to address the above risks. The risks prioritized according to their criticality will be tracked, monitored, and reported throughout the project lifecycle. The risk impact on key project processes will be analysed and the management will be notified of any important changes to the risk assessment.