

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

System Requirement Specifications (SRS) for TimeWise

Team 0

Lab group : SSP7

Date: February, 2020

Group members	Roles
Mantri Raghav	Project Manager
Dwivedee Lakshyajeet	Development Lead
Harding James	Back-End Developer
Alex Bernini	Front-End Developer
Mittal Madhav	Release Manager
Xue Xueting	QA Engineer
Koh Hui Ling	QA Engineer
Lek Jie Ling	QA Manager

Table of Contents

Problem Statement	4
Overview	4
Background	4
Overall Description	4
Investigation and Analysis Methodology	4
System Investigation	4
Analysis Methodology - Feasibility study and requirements elicitation	5
Analysis Methodology - System analysis and requirements elicitation	5
Analysis Methodology - Object-oriented design using UML	6
Analysis Methodology - Prototyping	7
Constraints	8
Scalability	8
Data and function mapping	8
Proprietary hardware and software	8
Batch updates vs. (close) Real-time updates	8
Project Schedule	8
Operational requirements	9
Help Desk Support	9
Application Services and Technical Support	9
Administration Features	9
System Interface independent of VRU	9
System hardware fail over and routine backup	9
Audit Trail	9
Functional Requirements	10
Input Requirements	11
Student account authentication and access	11
Tasks information	11
Process Requirements	12
MongoDB Transaction	12
Data integrity	12
Data validation	12
Performance	12
Data repository	12
Output Requirements	13
Transaction summary and confirmation	13

Exception Reports	13
Hardware Requirements	13
Network	13
Client Devices	13
Production Support Systems	13
Software Requirements	13
Client Operating Systems	13
Network System	13
Deployment Requirements	14

Problem Statement

There is no application provided by any service/company for NTU students to manage their timetable effectively. Current apps like NTU-Wave only allow the creation of a calendar using only the existing timetable generated by STARS and doesn't allow adding our own custom tasks/meetings. It is hence reduced to a timetable with only coursework related commitments. One cannot create tasks and to-dos due to which a student has to use another app to keep track of pending work/tasks, write it on paper or even go by memory! This is highly inefficient and can hamper the productivity of a student greatly. There is no one-stop application for keeping track of a student's schedule and allowing one to add custom tasks, not just coursework schedules parsed from the STARS website.

Overview

Background

Productivity is crucial to a student in college life. Organizing tasks and keeping oneself up-to-date with all the work and studies can help a student get good grades and keep life easy and organised in college. Unfortunately for now the current facilities - the apps are not sufficient to organize and add to-dos and tasks with a calendar.

As the number of classes and tasks add up for students, keeping track of it is getting to be tremendously tedious. Other productivity apps do not provide the facility to create a timetable after parsing it from STARS's website.

Overall Description

The system will allow a user to login using their university credentials and then their timetable will be scraped from the university website and added to our database. They will then be given a 7 day overview of lessons that they have enrolled in and will be able to add personal events to this overview.

Investigation & Analysis Methodology

System Investigation

TimeWise is being developed as a standalone utility app for NTU students with the aim of making daily and weekly planning of tasks and courses easier for students. The main users of this app will be NTU students. The existing planning apps will be researched thoroughly to understand their shortcomings. Most task planning apps don't contain the course schedule of the student which either forces the student to copy each class manually into the app or to use 2 different apps which is cumbersome. However, TimeWise will automatically download all course details and show it on the app, leaving the student with just the work of adding their own tasks. This will also be made easier by recommending 5 time slots for the student to do this task.

A cloud-based database and server will be used to store all important information. This will allow safe storage of data for all users.

Analysis Methodology

1. Feasibility study and requirements elicitation

Organize a development and implementation team composed of people knowledgeable about the current STARS planner and processes with which regular meetings will be held. A series of interviews with the managers and the developers of the STARS system and the U-wave app will be arranged. Interview and feedback from the personnel and staff working directly with general students is needed to define the current environment and future system requirements. A Feasibility and Risk Assessment study will be conducted to determine which solution(s) are most appropriate based upon the results of the interviews.

2. System analysis and requirements specification

An external view of the model of the TimeWise app including user authentication details, course schedule, and planned tasks will be developed using Unified Modeling Language (UML). This System Requirement Specifications documents will form part of the documentation for the project. Some desired features of the new system include:

- The ability to view tasks for the given week
- The ability add/edit/create events for a given day and time
- Import the students' timetable from the universities STARS system.
- Login and Logout
- Get time recommendations for new tasks

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

An external view of the model of the TimeWise app including user authentication details, course schedule, and planned tasks will be developed using Unified Modeling Language (UML). This System Requirement Specifications documents will form part of the documentation for the project. Some desired features of the new system include:

- The ability to view tasks for the given week
- The ability add/edit/create events for a given day and time
- Import the students' timetable from the universities STARS system.
- Login and Logout
- Get time recommendations for new tasks

b. Scopes and Limitations

Requirement analysis:

1. The application will be developed for Android and iOS smartphones and thus will require a touch screen. The output will be displayed on the screen as well
2. The application will allow the user to: Login using their NTU details, import their

class schedule, add daily tasks, see upcoming classes and tasks for the next 7 days, edit tasks (change name and time), delete tasks, log out, and update their class schedule.

3. The system will use JSON Web Tokens for secure storage of passwords in the MongoDB database.

Data analysis:

1. All data will be extracted through the Express server and stored into the MongoDB database.
2. Retrieval of data will be done through fetch instructions from the front-end to API endpoints
3. Data will be manipulated using Mongoose schemas with MongoDB
4. Data validation will be done using Express

Application architecture:

1. Information will be stored as Schemas in MongoDB. The schemas will be for the User, Task, and Schedule.
2. The application will be made in dark mode initially since this design is more pleasant for the user.
3. The application will be implemented using React Native for cross platform development.
4. The user interface has been designed using Figma.

3. Object-oriented design using UML

A detailed object-oriented design for creating and storing tasks will be developed. UML will be used again for the graphical representation and documentation of the design. The system will primarily concern itself with the timetable parsing process. At its core, a student will paste the timetable generated on STARS's website that will be processed in near real time. In addition, the system will allow students to add tasks. The system will be secured with a student's ID and password/PIN.

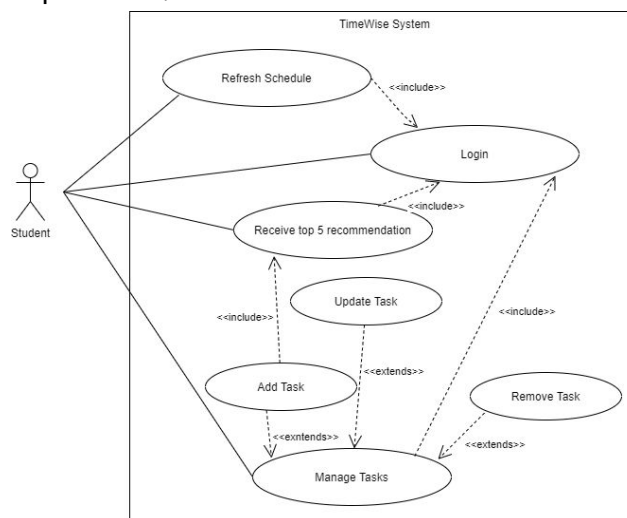


Fig. 1 Use Case Diagram for TimeWise

4. Prototyping

The online tool Figma will be used for implementing a limited and functional prototype for the planning system. The prototype will be a working example of the system for demonstration and proof of concept purposes only. It will include the end user interface. The prototype will be presented to the implementation team.

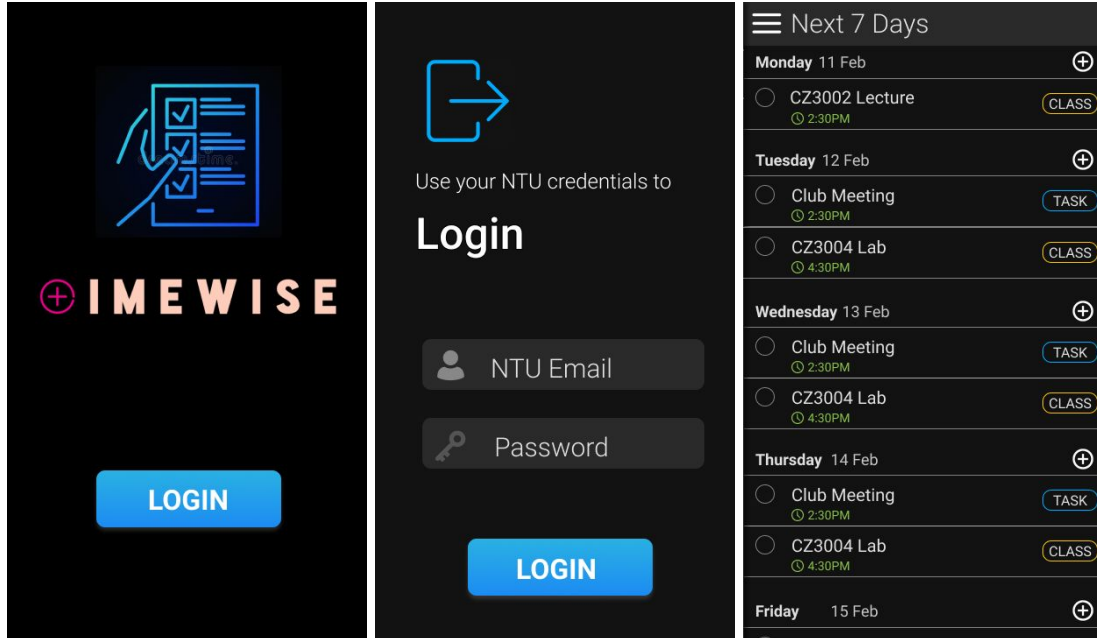


Fig. 2 Samples of Figma prototypes for the TimeWise application

Constraints

Scalability

The TimeWise app system scales well to increasing system demands. Error handling is limited to a few anticipated or common errors.

Data and Function Mapping

A task added to the app system can be readily used across devices using the same account. Also a new timetable added can be easily mapped to the existing app system.

Proprietary hardware and software

TimeWise requires Android 9.0 or iOS 11 to work on mobile devices. It has no other additional requirements.

Batch updates vs. (close) Real-time updates

There is no real-time update of registration system data for transactions through the TimeWise system.

Project Schedule

There is a three-month timeframe to implement a production system of the TimeWise app from project commencement in time for Fall 2020 registration.

Operational Requirements

Help Desk Support

System users will be able to send an email with any questions that are technical in nature, such as, slow or sluggish system response time, incompatible timetable format, application errors, system downtime inquiries, account lock-out assistance, etc. The email will be shown on the app listing page in the online stores.

Application Services and Technical support

Programmers and application developers will have access to source code to address bugs or system enhancements as deemed necessary. Network Administrator and database support is also required to maintain a 24x7 system uptime.

Administration Features

System security and access levels are provided in the online system. There are varying levels of system access and functional authority. Each student's access is limited to his/her own registration records. Only authorized system administrator(s) has access to all student timetables and task records. System access will be through MLab.

System Interface

The TimeWise system will remain operational and is independent of the usage of the app by other users. At any one time, multiple students may use the system for creating and editing timetables and tasks.

System hardware fail over and routine back up

Backend developers will handle system hardware tasks such as data back-up, hardware maintenance, fail over, scheduled system patches and maintenance.

Audit Trail

Heroku will keep logs of all transactions which have occurred, at what time, and which data was changed. The logs will be stored for 30 days before the cache is cleared.

Functional Requirements

The timewise system is a “timetable-planner” system that shall help the students to manage their activities and plan accordingly.

1. The student must be able to use the app for managing their tasks
 - 1.1. There must be a login system
 - 1.1.1. The login system must take the user’s NTU email address
 - 1.1.2. The login system must take the user’s NTU password
 - 1.1.3. The login system must scrape the user’s STARS planner using the entered details
 - 1.1.4. The login system must store the scraped data into MongoDB
 - 1.2. The application must have a task planner
 - 1.2.1. The task planner must allow the user to look at the details of their classes
 - 1.2.2. The task planner must allow the user to add a task
 - 1.2.2.1. The user must be allowed to choose the task name
 - 1.2.2.2. The user must be allowed to choose the task time
 - 1.2.2.2.1. The application must recommend 5 time slots in which the user can do the task
 - 1.2.2.2.1.1. The time slots must be 1 hour long
 - 1.2.2.2.1.2. The time slots must be unoccupied by another task
 - 1.2.3. The task planner must allow the user to edit tasks
 - 1.2.3.1. The user must be allowed to choose the task name
 - 1.2.3.2. The user must be allowed to choose the task time
 - 1.2.4. The task planner must allow the user to delete tasks
 - 1.2.5. The task planner must allow the user to complete a task
 - 1.3. The application must have a button to refresh the schedule
 - 1.3.1. The button must populate the app with just the classes which the student has for the current system
 - 1.4. The application must have a logout button

Input Requirements

Student account authentication and access

Every Student can use their school email and password to login to the system. This credential will serve as an identification purpose to their account and for future access. The email and password entered will be checked against the Student's school account, the checking will be done by the system in the back-end by logging in to the student's school account. The students will only be granted access to the TimeWise system only if the login to their school account is successful. After the first login of the student, the email and password will be stored in TimeWise database. This account will be disabled for the Students after graduation or separation from the university.

Tasks Information

Every student can add tasks according to their own schedule/timetable. They are required to enter task information such as task name and time of the task. This information will be set as a reminder to students so that they are able to view it and use it for future references. A notification reminder will also be sent to the students to remind them prior to task starting time entered.

Process Requirements

The following are among the inherent requirements that the timewise system must be able to handle.

MongoDB transaction

The system must be able to send, receive and trigger transactions to the MongoDB TimeWise database system.

Data Integrity

All processes must function as expected by the user without errors and the data processed and stored must be accurate.

Data validation

Login credentials must be validated before allowing the access to the system.

Performance

- Must handle concurrent usage of the system on a 24x7 basis.
- Send, receive and display user messages to assist the overall user experience.
- Must be able to add/update/delete data within 5 seconds.
- Send notification messages to students within 10minute before the starting of the task.

Data repository

The timewise system will maintain MongoDB TimeWise database as the main repository of data.

Output Requirements

Transaction summary and confirmation

Each user must be able to view their tasks for the week. MongoDB is being used to store the timetables. Successful committed transactions will be shown on Heroku.

Exception reports

System exception reports must be consolidated to record errors like scheduling conflict.

Hardware Requirements

Network

Internet access through any network

Client devices

Android 9.0 or iOS 11 compatible device

Production support systems

Heroku Dyno support (back-up storage, redundant drives, UPS, etc.)

Software Requirements

Client Operating Systems

- iOS 11
- Android 9.0

Network system

Network software and protocols in order for systems to communicate:

- TCP/IP
- HTTP

Deployment Requirements

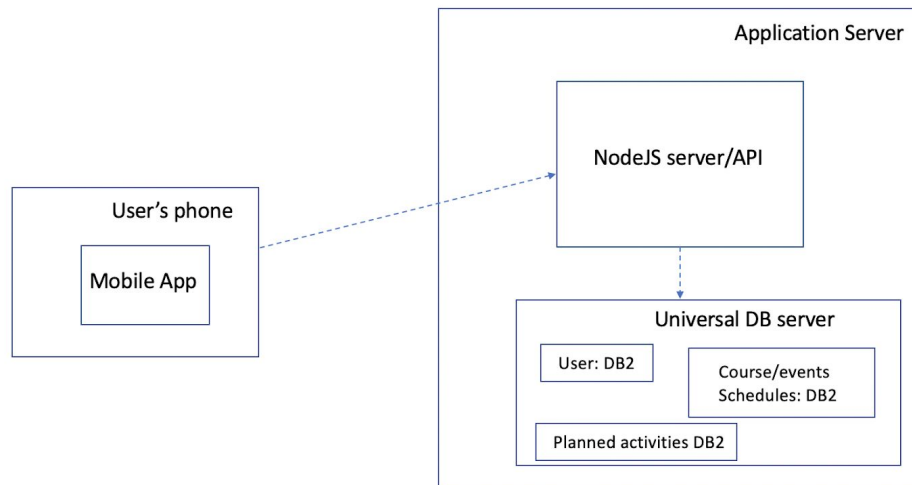


Fig. 3 Deployment Requirements of the TimeWise application

The frontend which is a mobile app is to be deployed on iOS and Playstore using expo. The backend which is an API server which the app consumes data from and a database would be deployed on a Heroku Dyno instance with a Linux environment. The data will be stored and received from a MongoDB database deployed on MLab.