

Software Requirements Specification For Dwebble

Group 12
2020.02.23
Version 1

Prepared by: Ibrahim AbuAnz
Reviewer and Editor: Elias Julian Marko Garcia

Table of Contents

REVISION HISTORY.....	4
1. INTRODUCTION.....	5
1.1 Overview.....	5
1.2 Goals and Objectives.....	5
1.3 Scope.....	6
1.4 Definitions.....	6
1.5 Assumptions.....	7
2. GENERAL DESIGN CONSTRAINTS.....	7
2.1 Dwebble Application Environment.....	7
2.2 User Characteristics.....	8
2.3 Mandated Constraints.....	9
2.4 Potential System Evolution.....	9
3. NONFUNCTIONAL REQUIREMENTS.....	9
3.1 Usability Requirements.....	9
3.2 Operational Requirements.....	9
3.3 Performance Requirements.....	10
3.4 Security Requirements.....	10
3.5 Safety Requirements.....	10
3.6 Legal Requirements.....	10
3.7 Other Quality Attributes.....	10
3.8 Documentation and Training.....	11
3.9 External Interface.....	11
3.9.1 User Interface.....	11
3.9.2 Software Interface.....	11
4. SYSTEM FEATURES.....	12
4.1 Feature: Setting Rules.....	12
4.1.1 Description and Priority.....	12
4.1.2 Use Case: Set Rules.....	12
4.1.3 Additional Requirements.....	12
4.2 Feature: Adding Class/Professor/Room Details.....	13
4.2.1 Description and Priority.....	13
4.2.2 Use Case: Provide Class/Professor/Room Details.....	13
4.2.3 Additional Requirements.....	13
4.3 Feature: Generating Schedule.....	13
4.3.1 Description and Priority.....	13

4.3.2 Use Case: Generate Schedule.....	14
4.3.3 Additional Requirements.....	14
4.4 Feature: Downloading Schedule.....	14
4.4.1 Description and Priority.....	14
4.4.2 Use Case: Download Schedule.....	15
4.4.3 Additional Requirements.....	15

Revision History

Version	Date	Name	Description
0.1	2020.02.23	Ibrahim AbuAnz	Initial Document
0.2	2020.02.29	Elias Garcia	Editing and review
1.0	2020.02.29	Elias Garcia	Version 1 release

1. Introduction

1.1 Overview

The document aims at presenting requirements specifications for the Dwebble Class Scheduler application, **hereon simply “Dwebble”**, for the university. The application is likely to be accessed as a web application that can be accessed from a desktop as well as mobile devices. It will be used by the administration and professors of the university internally to set different rules for teaching classes and to generate schedules based on those rules and other requirements dynamically. The document contains detailed information about the functional and non-functional system requirements and high-level design constraints for the Dwebble application. The interactive Dwebble application will help to make rules and constraints from end-users or administrators and will generate one or more possible schedules based on the current set of rules and constraints provided to it. The application will help in saving time and effort in schedule preparation for an educational institution.

This document is only for requirements specification. No details about the system design, analysis, and development, and no information about the project management aspects in the development and deployment of the Dwebble application is a part of this document.

The intended audience of the UMKC CSEE department and administration staff who want to save their time and prevent unnecessary manual effort from handling the complexity and workload of schedule generation for courses in each semester. Such stakeholders can check this document to understand how this system will work and what kind of functional and facilities it will provide to its end-users.

1.2 Goals and Objectives

The primary goals of the Dwebble application are,

1. Allowing administrators to set rules and constraints for class schedules, teachers preferences, class timing and availability of rooms
2. Generating one or more schedules dynamically in real-time based on the rules, constraints, and preferences provided by an administrator
3. Saving and storing schedule related information, such as previously entered staff schedules and restrictions, for later retrieval.

4. Allowing professors to access their class schedules generated by the system
5. Automation of scheduling as an activity: there should not be significant intervention on the part of administrators.
6. Offering a mechanism to the end-users to select a schedule based on different weights calculated for each schedule so that a user can find the best-suitable schedule based on a circumstance, i.e. in the case of the application not finding a non-conflicting schedule to use for a given set of inputs.

The main objective of the application is to make class schedule planning and generation faster and simpler for the end-users by handling the complex tasks to prepare a schedule based on different sets of changing rules, constraints, and preferences time-to-time

1.3 Scope

The scope of the Dwebble Application is limited to set rules and to generate schedules only. It will not cover other aspects of class and course management, student information management, or other business processes of a university.

There are two main components of the system, those are administration and generation. The scope of the administration component is limited to setting rules, configurations, and adding information about classes, professors, availability of rooms, class timing, preferences of professors on taking different classes from a semester to another. All such information would be fed into the system through a definite format to reduce the chances of ambiguity and to restrict errors in the system. The second component, generation, will automatically generate a schedule based on the information provided through the administration component. All generated schedules would be provided to end-users to select a suitable one or to make further modification through the administration component.

1.4 Definitions and Conventions

1.4.1 Definitions

Dwebble Application – Web-based application to generate class schedules automatically based on rules, setting and configurations provided to it dynamically in real-time.

Use case – describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.

Scenario – one path through a use case

Actor – user or another software system that receives value from a use case.

Role – a category of users that share similar characteristics.

Project – activities that will lead to the production of the product described here. Project issues are described in a separate project plan.

Shall – adverb used to indicate importance; indicates the requirement is mandatory. “Must” and “will” are synonyms for “shall”.

Should – adverb used to indicate importance; indicates the requirement is desired but not mandatory.

1.4.2 Conventions

Any section, sub-section, or item not completed will be marked simply with **TBD** – To Be Determined – and will be completed at a later date when more information and knowledge of the problem domain is obtained.

1.5 Assumptions

It has been assumed the web-based Dwebble application will follow distributed processing to speed-up data processing. It can be accessed by multiple users at a time. However, there will be restricted on accessing different types of data in the system by different types of users based on their roles concerning the system. Finally, it is assumed that end-users will have access to a computer with a modern web-browser and either internet access or internal network access such that Dwebble is accessible.

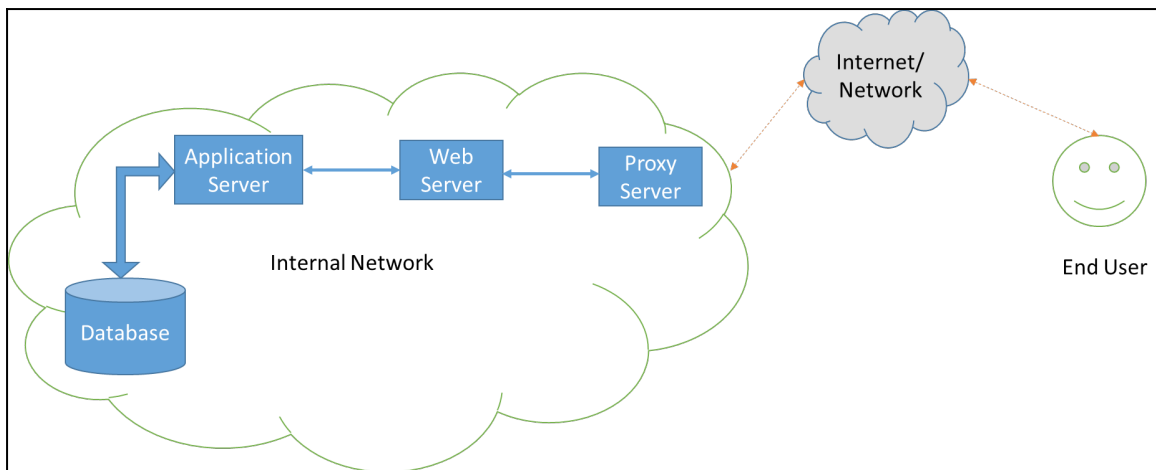
2. General Design Constraints

2.1 Dwebble Application Environment

The Dwebble application will work on a distributed network connecting to a proxy server, web server, and a database server. The end-users will access the system through suitable web interfaces designed for the system. The client requests from those web interfaces will come through the proxy server and web server to ensure security in accessing the system over the Internet and the internal network. The

application server hosting the Dwebble application will handle the request and may require establishing a connection to the back-end database server. Based on the processing of data collected from the database server, it will generate a response for the client's query and will send back the response to the client through the proxy and web server and networks.

The database server will provide data about the professors, rooms, rules, constraints and so on. It will be accessed through a standard ODBC/CLI interface. There will be security rules and access controls implemented in the Dwebble application. So, it will require end-users to go through a user authentication process to access data. All data will not be accessible by all types of end-users. For example, an administrator will be responsible to set the rules and configurations. Professors will not be allowed to make changes to those rules. They will be allowed to enter their preferences on taking various classes. There will be different interfaces for different types of end-users.



2.2 User Characteristics

There will be two different end-user groups for the Dwebble application. A person can take both roles for two different end-user groups. For example, a user who is an administrator can also take the role of a professor but not vice versa.

The first type of end-users is one or more administrators. An administrator will have authority and more control over the system. An administrator will interact mainly with the administration component of the system. Usually, the principal or head of the department can take the role of an administrator to prepare class schedules. An administrator is responsible to set rules, enter data about classes,

professors, rooms, and other settings and configurations. They can also enter data about constraints.

Other groups of end-users are the professors. They are responsible to provide details of their preferences on taking classes in terms of class-timing and willingness to teach a task on a priority basis. For example, a professor may give preference on taking class A over class B. While making a schedule the system will follow the preference in the best possible way under the given set of rules and other data.

All end-users can check schedules and to select on schedules. However, an administrator will get all prioritized schedules generated by the system and can select one. However, individual professors will get only the selected schedule as the final one. They won't be able to make any changes to the schedule, but an administrator can make changes to a schedule.

All end-users must be proficient in using web interfaces, and computers or mobile devices to access the system.

2.3 Mandated Constraints

The Dwebble application will be a web application that can be accessed through a web interface on a desktop or a mobile device irrespective of the underlying operating system of the device. However, it will require connectivity to the Internet or the internal network hosting the system, to access the system.

2.4 Potential System Evolution

No changes to the primary goals and objectives to the system is allowed in future. The main functionalities of the system will remain unchanged in future. However, usability and accessibility aspects may change in future based on the evolution and changes in underlying technologies.

3. Nonfunctional Requirements

Non-functional requirements should be fulfilled by the system to support the implementation of the functional requirements.

Other non-functional requirements as specified by stakeholders include:

- **Usability** – the application must be easy to use, particularly for those who are considered generally “not tech savvy”
- **Simple design** – following the usability NFC, the design of the application should not be over-engineered. It should be basic and achieve its job without confusing the user in the process.

- **Low technical overhead** – the system should be easy to maintain, host, and update as necessary.

3.1 Usability Requirements

The system will follow different standards and measurements to meet the usability requirements of the system. The key usability requirements are,

- The system should be easy to use by its end-users
- The system should allow end-users to come out from an error easily by following on-screen instructions and should not make the end-users feel stuck
- The end-users should be able to memorize the functions of the system and how to perform those intuitively
- The system should ask for minimal interactions from the end-users
- An end-user should be able to use the system properly within 2 to 3 days of training

3.2 Operational Requirements

The system should not allow end-users to set ambiguous rules by restricting their interactions to the system by using close-ended queries on input forms. The performance of the system should not depend on the local hardware or software capacity of an end user's device.

3.3 Performance Requirements

The system should be available 24x7 without significant downtime. It should allow up to 100, non-administrative, concurrent users but when a schedule preparation is in progress then it should be atomic, and no changes should be allowed in between. It should take no longer than 5 to 10 seconds to make a schedule if possible. If it finds no schedule is possible then it should respond the same to the user by showing the reasons within a maximum of 5 seconds.

3.4 Security Requirements

- It should have user authentication implemented in the system. Users should provide valid user ID and passwords to access the system. Based on the provided details, the system should be able to determine the role of an end-user and should allow access to the system accordingly.

- It requires a secure connection between an end-user and the system so, there should be a time-out limit for the system when an end-user remain inactive for more than 10 minutes in a session. When a time-out occurs, the system should log out automatically from the user account and should ask for re-login.

3.5 Safety Requirements

To remain safe in the operating environment, the servers should be protected from possible hazards and data losses. Encoding should be there to ensure secure data transmission for sensitive information over the Internet.

3.6 Legal Requirements

Personal data about professors, their preferences in taking classes should be protected from the rest of the end-users other than the administrators. The system should follow all applicable legal standards and regulations for data privacy, protection, and intellectual property rights protection.

3.7 Other Quality Attributes

The system should be available 24x7 without significant downtime for maintenance. The total maintenance time in a week should be lesser than 2% of the total up-time.

3.8 Documentation and Training

The Dwebble application will have on-screen support and help available for end-users. Hence, no hard copy for training documentation will be provided. However, there will be short training videos, and user manual available online with the system for its end-users. The application requires no installation at a local computing device. All data processing will be performed online on a distributed network and servers.

3.9 External Interface

3.9.1 User Interface

The user interface for an administrator is likely to have more complexity than the user interface for a professor who is not an administrator. Hence, an administrator should be able to understand at least 50% of the user interface without training and any difficulty in a maximum of 1 hour. A professor should be able to understand at least 80% of their user interface in first use without any training and

difficulty. And the level of understanding should increase with subsequent usage.

The user interfaces will be intuitive, and aesthetically suitable for the targeted audience. The use of the color scheme, typography and design should be minimalistic and focused on its functionalities rather than the design aspects.

The interfaces should be responsive so that it is adapted for most of the screen sizes automatically without affecting user experiences of end-users.

Online or in-person support will be provided for the end-users. Administrators can learn, via video-chat sessions or in-person meetings, with the development team for training to learn how to set the rules, settings, and configurations. Training for a maximum 1 day should be enough to use the system.

There will be input forms asking for details through drop-down menus, radio buttons, checkboxes, interactive calendars, clocks, and text boxes. Pop-up screens will be there to provide system responses and updates.

3.9.2 Software Interface

The client-side and server-side of the web application will communicate using HTTPS and JSON.

4. System Features

4.1 Feature: Setting Rules

4.1.1 Description and Priority

An administrator will be responsible to set the rules in the system to make a schedule. Rules can have different priority and weights. Based on those a schedule is generated, and a schedule also has a calculated weightage. This is a high priority feature.

The cost to implement this feature is high as this is one of the important features of the system. The risk is high as an administrator may set an ambiguous set of rules that cannot generate a schedule, in that case, the system's effort will be wasted. And the value is also high as it determines the cost of schedule generation. An error in setting rules may cost more in terms of system resources, time, and effort.

Estimates:

- Cost: High
- Risk: High
- Value: high

4.1.2 Use Case: Set Rules

Title	Set Rules
Basic Path	<ol style="list-style-type: none"> 1. Administrator mentions rules in a form 2. Administrator submits the rules for a schedule 3. The system checks whether all rules are distinct or not. 4. If all rules are distinct, the system saves the rules
Alternate Path	<ol style="list-style-type: none"> 1. Administrator mentions rules in a form 2. Administrator submits the rules for a schedule 3. The system checks whether all rules are distinct or not. 4. If the system finds two submitted rules are identical for a schedule, it asks the administrator to modify those rules. 5. The administrator may change those or may drop the set of rules.

4.1.3 Additional Requirements

An administrator must be logged in to the system with valid details.

4.2 Feature: Adding Class/Professor/Room Details**4.2.1 Description and Priority**

An administrator can add details of a class and professors. This is also a high priority feature.

The cost is high as this is a high priority feature. The risk is low as class and professor data is already available in the underlying database. So, there are fewer chances of errors in entering data. Value is high as it will help in creating schedules.

Estimates:

- Cost: High
- Risk: Low
- Value: High

4.2.2 Use Case: Provide Class/Professor/Room Details

Title	Provide Class/Professor/Room Details
-------	--------------------------------------

Basic Path	<ol style="list-style-type: none"> 1. Administrator selects classes, professors 2. The system retrieves the professor's preferences 3. The system retrieves class timing details and room availability details 4. System asks for confirmation from the administrator 5. The administrator checks and confirms the details 6. The system saves a copy of all provided details for a schedule
Alternate Path	<ol style="list-style-type: none"> 1. Administrator selects classes, professors 2. The system fails to retrieve the professor's preferences 3. System fails retrieves class timing details and room availability details 4. The system asks the administrator to enter those details 5. The administrator may enter those details or cancels the process

4.2.3 Additional Requirements

An administrator must be logged in to the system with valid details.

4.3 Feature: Generating Schedule

4.3.1 Description and Priority

An administrator asks the system to generate a schedule by providing all relevant details. It costs high as this is the primary feature of the system and it consumes resources to computer schedules. The risk is low as the system will either provide at least one schedule or prove a report on why no schedule generation is possible. The value is very high as this will give the primary output of the system.

Estimates:

- Cost: High
- Risk: Low
- Value: High

4.3.2 Use Case: Generate Schedule

Title	Generate Schedule
Basic Path	<ol style="list-style-type: none"> 1. An administrator asks the system to generate the schedule after providing all details 2. The system will process data to generate the schedule 3. The system will provide a list of possible schedules to the administrator along with weights of the

	schedules 4. An administrator selects one schedule from the list and confirms it 5. The system notifies and sends the selected schedule to all of the professors.
Alternate Path	1. An administrator asks the system to generate the schedule after providing all details 2. The system will process data to generate the schedule 3. The system finds logical conflicting rules that make impossible to generate a schedule 4. The system provides a report on the conflict to the administrator and asks to change rules 5. The administrator may make changes or cancel the process or may save it for later.

4.3.3 Additional Requirements

An administrator must be logged in to the system with valid details.

4.4 Feature: Downloading Schedule

4.4.1 Description and Priority

When a professor gets a notification about a schedule, the professor must be able to download it and check it. The cost is low as it is not the primary feature of the system. There is no risk involved as the schedule is already generated by the system and no user input is required. The value is high as professors will use it.

Estimates:

- Cost: Low
- Risk: Low
- Value: High

4.4.2 Use Case: Download Schedule

Title	Download Schedule
Basic Path	1. Professor will request download of a schedule notified to the professor 2. The system will process the request by preparing a copy of the schedule for the professor only by omitting the details of the schedule of other professors. 3. Professor gets the downloaded copy at the local device.

Alternate Path	N/A
-----------------------	-----

4.4.3 Additional Requirements

A professor must be able to login to the system before downloading.