Boardzilla - Software Requirements Specification

Matthew Paulin	Hargun Bedi	Dylan Smith	Chenwei Song
paulinm	bedih	smithd35	songc12
400187147	400185463	001314410	400124879

Tianzheng Mai
mait6
400143042

February 2, 2021

Contents

1 Introduction

This section of the SRS should provide an overview of the entire SRS.

Boardzilla

widgets

- weather
- calendar
- sticky notes
- news (google news api)
- stock prices

Functionality

- Momentum-like welcome msg
- plus sign to add new widgets
- canvas starts blank, prompted with instructions to add first widget
- login to save data and widgets

Implementation

- React, JS, HTML5, CSS3
- bulma, materialize, or bootstrap for base styling
- hosting: AWS
- domain: ?
- backend:
- db: MongoDB / other noSQL
-

1.1 Purpose

This document will elucidate the requirements necessary to create Boardzilla. Some system requirements have been gathered from the project outline and others have been generated to create an acceptable final result. Additionally, this document will contain an explanation of the project and its constraints, the stakeholders of the project, as well as a quantitative measure of the desired finished product. The intended audience for this document include the team members who are developing the product, the Professor, Dr. Khedri, as well as any teaching assistants who will be grading this document.

1.2 Scope

The product being developed is called Boardzilla, an online application that will serve as a dashboard for a variety of pertinent information. Each user can have a unique dashboard containing their desired widgets from the available options. These options include: a weather widget, a sticky notes widget, a calendar widget, a news widget, and an widget showing live stock prices. This application will serve as a daily briefing to users, presenting live, relevant, and customizable information that is all aggregated on a single page.

1.3 Definitions, Acronyms, and Abbreviations

- **Dashboard**: A user interface or web page that gives a current summary, usually in graphic, easy-to-read form, of key information relating to progress and performance, especially of a business or website.
- **Widget**: An element of a user interface that displays information or provides a specific way for a user to interact with an application.

1.4 References

1.5 Overview

The remainder of this document will provide an overall description of our product, a use case diagram depicting the business events, the functional and non-functional requirements, and the division of labour.

2 Overall Description

2.1 Product Perspective

Our system will provide users with a large variety of widgets to customize their dashboard. Similar dashboard applications are focused on relaying analytical information such as team or product statistics. Boardzilla's widgets provide users with options relevant to broader demographics rather than individual businesses. These widgets can then be combined and configured by the user, providing them with a dashboard that is unique to them. The standalone versions of the widgets such as Google News or The Weather Network provide the same information but are lacking in convenience due to being largely isolated applications. The system will interface with external databases that contain news data, weather data, stock data, and calendar data.

2.2 Product Functions

When a user navigates to Boardzilla, they will be presented with a login page. If they are unregistered, they will have to first register before being able to access their personalized dashboard. Once logged in, a user will be able to see a welcome page greeting the user personally with a friendly message. The app will also display the user's widgets in their chosen configuration. By dragging the widgets around the screen, the positioning and layering can be adjusted to suit the user's needs. Each widget will have an ability for minimization or deletion and there will also be a menu to add additional widgets. The available widgets will include weather, stock, calendar, news and sticky notes. When adding new widgets, the user must select specific widget options if necessary. All of the user's dashboard and individual widget customization will be saved in the cloud and accessible through their account on multiple machines.

2.3 User Characteristics

Boardzilla will be designed to appeal to everyone in the English speaking world, the only requisite knowledge being how to use a computer and web browser. To that end, intended users of Boardzilla will not be required to have any other technical background or education besides an intermediate understanding of the English language. Additionally, prior experience with our application will not be required because all relevant information will be provided to the user upon registration.

2.4 Constraints

- Boardzilla must be given internet access to fetch data from external databases in order to populate the widgets with data.
- Boardzilla must obey local laws and regulations.
- Boardzilla must follow the terms of service of all external APIs.

2.5 Assumptions and Dependencies

Assumptions:

- Users will know English or have a browser capable of translating to their spoken language.

Dependencies:

- Google API – expand

a) List each of the factors that affect the requirements stated in the SRS

b) These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS

    - **Example**: An assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

2.6 Apportioning of Requirements

In future versions we will:

- Enable the user to choose from a wider variety of widgets.
- Add multiple language and region support for the text within the application.
- Increase the level of accessibility for physically or mentally impaired users.
- Allow users to see previously retrieved information in an offline version of the application.
- Allow users to save and access multiple layout configurations.

3 Use Case Diagram

This section should provide a use case diagram for your application.

a) Each use case appearing in the diagram should be accompanied by a text description.

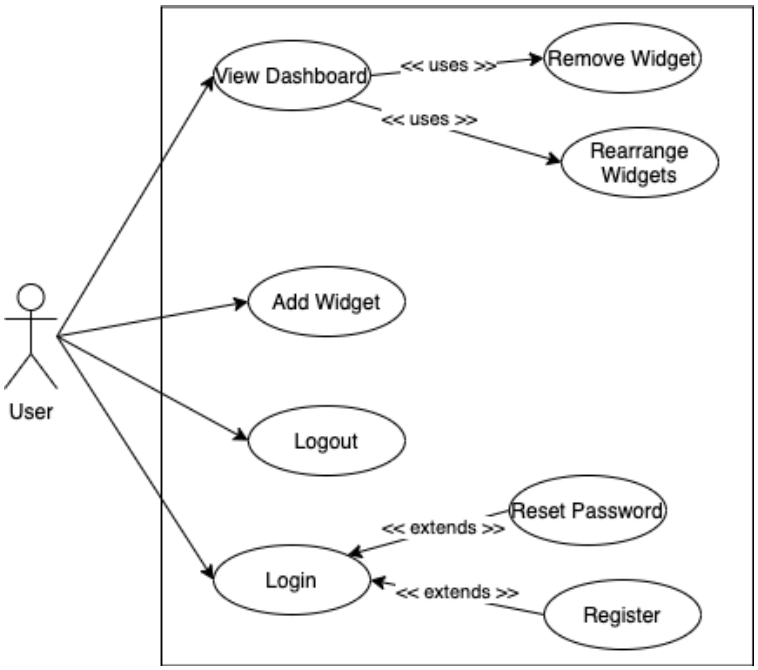

Figure 1: Use Case Diagram

4 Functional Requirements

VP1 User

BE1 User wants to view their dashboard

i. The system will check if the user is logged in to their account with the system.
ii. If the user is not logged in, the system will redirect the user to the system registration page.
iii. The system will fetch the user's widgets from the internal database.
iv. The system will display the user's widgets in accordance with the user's chosen positioning.
v. The system will display the add and remove widget options.
vi. The system will update the widgets with current information.
vii. The users will be able to access more information by hovering over the associated help icons.

BE2 User registers for an account

i. The system will provide the user with a sign up form.
ii. The system will verify the validity of the information and will create an account in the database if the information is valid.
iii. The system will display an error message upon an invalid registration.
iv. The system will redirect the user to a newly created, empty dashboard and will display system instructions.

BE3 User attempts to login to the system

i. The system will display a login form.
ii. The system will verify that correct login information was provided.
iii. The system will redirect the user to their own dashboard.
iv. The system will display an error message after an invalid login attempt.

BE4 User wants to add a widget to their dashboard

i. The system will provide the user with a list of possible widgets.
ii. The system will allow the user to input a widget selection and will redirect them to the setup page for that widget.
iii. After widget configuration, the system will prompt the user to place the widget on their dashboard.
iv. The system will display an error message indicating any incorrect widget parameters.
v. The system will fetch the widget data.
vi. The widget and its parameters will be added to the system's database.

BE5 User wants to remove a widget from their dashboard

i. The system will provide the user with a list of existing widgets to delete.
ii. The system will remove the selected widget from the database.
iii. The system will remove the selected widget from the dashboard

BE6 User wants to log out of the application

i. The system will invalidate the user's session.
ii. The user will be redirected to the login page.

BE7 User wants to reset their password

i. The system will send a password reset email containing a temporary password.
ii. The user will be redirected to the change password page where they can modify their password.

BE8 User wants to modify their dashboard layout

i. The positioning of the widgets will be updated in the system's database.
ii. The display will reflect the updated widget positioning.

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF1. The system shall use modern styling whenever possible.

LF2. The system shall be visually appealing to 77% of users.

5.1.2 Style Requirements

LF1. The text size and font must be legible at a normal computer viewing distance.

LF2. The fonts, colours, and icons must remain consistent throughout the system.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH1. Any buttons must be large enough to be located and pressed within 5 seconds.

UH2. The users must be presented with general instructions when logging in for the first time.

UH3. The language should be understandable to anyone with a fifth grade level of reading comprehension.

5.2.2 Personalization and Internationalization Requirements

UH1. Boardzilla must allow users to adjust the positioning and layering of the widgets on their personal dashboard.

UH2. Widgets must accept valid parameters from users.

5.2.3 Learning Requirements

UH1. The user interface must be understood by a user within 10 minutes of system use.

5.2.4 Understandability and Politeness Requirements

UH1. The symbols must convey the correct meaning to users 95% of the time.

UH2. The language in the app must be grammatically correct 99% of the time.

5.2.5 Accessibility Requirements

UH1. Any adjacent colours used within the user interface must appear distinct to users afflicted with colour blindness.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR1. The system shall respond to network requests with the necessary information in less than 3 seconds 99% of the time.

PR2. A request to fetch user data from the external **APIs** must be sent in under 10 seconds 99% of the time.

PR3.

5.3.2 Safety-Critical Requirements

PR1. N/A

5.3.3 Precision or Accuracy Requirements

PR1. The system must show correct widget information 99% of the time.

PR2. The database must store current information 99.9% of the time.

5.3.4 Reliability and Availability Requirements

PR1. Boardzilla shall be available for use 99.99% of the time.

PR2. The system shall require internet access to function.

5.3.5 Robustness or Fault-Tolerance Requirements

PR1. Boardzilla must display data consistently across supported devices.

PR2. The system shall remain visible, but with stale data in the case of internet loss.

PR3. The system shall notify active users in the case of a system failure, or loss of connectivity to the system.

5.3.6 Capacity Requirements

PR1. The system shall be able to handle up to 100 users concurrently.

5.3.7 Scalability or Extensibility Requirements

PR1. The system shall be designed such that additional widgets can be added in future versions without modifying more than 10% of existing software.

5.3.8 Longevity Requirements

PR1. The system shall remain online for at least 2 years after software delivery.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE1. Boardzilla shall be operable in any physical environment that mobile phones or desktop computers can be operated in.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE1. The application shall interface with APIs that provide widget data at no cost.

5.4.3 Productization Requirements

OE1. N/A

5.4.4 Release Requirements

OE1. The system must have 90% of software issues patched before release.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS1. System updates shall not keep the system down for longer than 10 minutes.

MS2. The system shall have an uptime of 99.99%.

5.5.2 Supportability Requirements

MS1. The system's instructions shall explain 99% of scenarios.

5.5.3 Adaptability Requirements

MS1. N/A

5.6 Security Requirements

5.6.1 Access Requirements

SR1. The system shall only allow access if a valid username and password are entered.

SR2. The system shall only provide a user with access to their dashboard.

5.6.2 Integrity Requirements

SR1. The system must sanitize all data that is collected from users.

SR2. The system must employ rate limiting to cap the addition of widgets to one widget per twenty seconds per user.

SR3. The user must not be able to manipulate other user's data.

5.6.3 Privacy Requirements

SR1. The system must not release user information to any third party.

SR2. The system shall not store unencrypted user passwords.

SR3. The system shall not collect any unnecessary user data.

5.6.4 Audit Requirements

SR1. After a security audit, at least 90% of the changes should be made within a month.

5.6.5 Immunity Requirements

SR1. Software security patches must be installed within a week of their release.

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP1. The iconography and language contained within Boardzilla will be inoffensive to 99.9% of users.

5.7.2 Political Requirements

CP1. N/A

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR1. The system must obey local laws and regulations.

LR2. Boardzilla must follow the terms of service of all external APIs.

5.8.2 Standards Requirements

LR1. N/A

A Division of Labour

Team Member	**Contributions**
Matthew Paulin	Add contributions here
Hargun Bedi	Add contributions here
Dylan Smith	Add contributions here
Chenwei Song	Add contributions here
Tianzheng Mai	Add contributions here

Table 1: Division of Labour

A.1 Signatures

Matthew Paulin Date

Hargun Bedi Date

Dylan Smith Date

Chenwei Song Date

Tianzheng Mai Date

IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
  - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or underline
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.