

SOFTWARE REQUIREMENTS SPECIFICATION (SRS) FOR

Team Name: Java The Hutts

Class: Fall 2023 - Comp 490/L

Instructor: Edmund Dantes

Revision History

Revision Letter	By	Change Description	Date
1.0	Entire group	Initial requirement spec	12 Nov 2023

Table of Contents

	Page
1. INTRODUCTION	1
1.1 Scope	1
1.2 Product Value	1
1.3 Intended Audience	1
1.4 Intended Use	1
2. FUNCTIONAL REQUIREMENTS	2
3. EXTERNAL INTERFACE REQUIREMENTS	3
3.1 User Interface Requirements	3
3.2 Hardware Interface Requirements	3
3.3 Software Interface Requirements	3
3.4 Communication Interface Requirements	3
4. NON FUNCTIONAL REQUIREMENTS	3
4.1 Security	3
4.2 Capacity	3
4.3 Compatibility	3
4.4 Reliability	4
4.5 Scalability	4
4.6 Usability	4
4.7 Other	4
5. QUALIFICATION PROVISIONS	5
6. NOTES	8
6.1 Acronyms and Abbreviations	8

Table of Figures

Page

NO TABLE OF FIGURES ENTRIES FOUND.

List of Tables

Page

TABLE IV. REQUIREMENTS VERIFICATION	5
TABLE V. XYZ SRS TO ABC SYSTEM SPECIFICATION REQUIREMENTS TRACE	6
TABLE VI. ABC SYSTEM SPECIFICATION TO XYZ SRS REQUIREMENTS TRACE	6
TABLE VII. ACRONYMS AND ABBREVIATIONS	7

1. INTRODUCTION

1.1 Scope

This document defines the software requirements for Java The Hutts, Project Thisplay: a simple, programmable 'smart' display and companion iOS and web apps to interact/publish content to the display.

1.2 Product Value

The value of this project ranges from desk/geek toys (e.g. use as a simple digital photo frame), use as a dynamic information 'ticker' (e.g. to display smart home sensor(s) state, calendar events, weather information, etc.) to on-demand advertising use.

1.3 Intended Audience

The intended audience will be largely aimed at tech 'tinkerers' or 'makers' who want a quick and easy solution to create custom/programmable 'smart' displays/signage.

1.4 Intended Use

As with the product value, users will largely find value via the accessibility and dynamic nature of the display. Instead of needing a full computer, tablet, or secondary phone, users will be able to display custom text and images on multiple small displays, at low cost, and place them wherever they want.

2. FUNCTIONAL REQUIREMENTS

2.1 Requirements for the hardware firmware (FUNC_RPI_XXX):

FUNC_RPI_001: The firmware shall have the ability to output to e-ink, and lcd display panels.

FUNC_RPI_002: The firmware shall support caching of user data.

FUNC_RPI_003: The firmware shall support connecting to the web and ios applications over bluetooth.

FUNC_RPI_004: The firmware shall support connecting to the web and ios applications over local wifi/networking.

FUNC_RPI_005: The firmware shall accept image files from connected web and iOS applications for display on the connected panel.

FUNC_RPI_006: The firmware shall accept raw text input from connected web and iOS applications for display on the connected panel.

FUNC_RPI_007: The firmware shall be able to rotate the information being displayed after user specified time intervals.

2.2 Requirements shared by the Web and iOS applications (FUNC_COM_XXX):

FUNC_COM_001: The applications shall use standard HTTP/S protocol when connecting to the display via local network.

FUNC_COM_002: The applications shall use BLE when connecting to the display via direct/Bluetooth.

FUNC_COM_003: The applications shall allow users to browse and select local jpeg or png image files for display on the device.

FUNC_COM_004: The applications shall allow users to input text directly to be displayed on the device.

FUNC_COM_005: The applications shall allow users to configure a specific amount of time that an image or block of text should be displayed for.

2.3 Requirements unique to the Web application (FUNC_WEB_XXX):

FUNC_WEB_001: The web application shall detect the browser it is running in, and display an error message on non-chromium (unsupported browsers.)

2.4 Requirements unique to the iOS application (FUNC_IOS_XXX):

FUNC_IOS_001: The iOS application shall check for and notify the user if bluetooth or network permissions have not been granted

FUNC_IOS_002: The iOS application shall have a list of saved/previous connected devices, which can be tapped for quickly reconnecting to a previously used device.

FUNC_IOS_003: The iOS application shall allow users to manually configure a new display/device to connect to.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interface Requirements

EXTINTF_SRS_001: The web and iOS applications shall adhere to the following shared style guides:

1. Use a minimalistic design with a clear, sans-serif font for easy readability.
2. Be compatible with both desktop and mobile browsers.
3. Provide visual feedback upon submission, such as a confirmation message or animation.
4. Use native/system provided controls wherever possible.

EXTINTF_SRS_002: In addition to the above shared guidelines, the web application shall provide a responsive web interface accessible via standard web browsers. The interface shall include a toggle to switch between text and image mode. In the text mode, the interface will display a text input field with the current character count entered, along with reset and submit buttons. Upon pressing the submit button, the text shall be transmitted to the Raspberry Pi 4 and displayed on the attached screen.

1. The text input mode shall include:
 - 1.1. A responsive, multi-line text input field, that expands to fill the width of the page.
 - 1.2. A live character count indicating the current number of entered characters, and the total/maximum number of allowable characters.
 - 1.3. A reset button to clear all entered characters from the input field.
 - 1.4. A submit button to transmit the contents of the input field to the Raspberry Pi/display device.
2. The image mode shall include:
 - 2.1. An upload/browse button to allow users to select a local image file.
 - 2.2. A preview of the chosen image.
 - 2.3. A button to convert the image color format if using an e-ink display.

EXTINTF_SRS_003: The iOS application shall follow the same guidelines shared above, while adhering to standard iOS conventions:

1. The application shall utilize standard iOS tab-bar for switching between text and image modes.
2. The application shall use the standard iOS user defaults api to store session data and saved device(s) information.

3.2 Hardware Interface Requirements

EXTINTF_SRS_004: The firmware shall be compatible with the following hardware:

1. Raspberry Pi 4 Model B and Raspberry Pi Pico W connected to a display panel for text/image output.
 - 1.1. The display may be of LCD or e-ink (with appropriate driver IC.)
 - 1.2. The display connection may be via HDMI, or direct SPI/GPIO pins.
2. The system shall require a network connection, which can be through Ethernet or Wi-Fi, adhering to the IEEE 802.11 standards.
3. Communication with the web interface shall use the HTTP/HTTPS protocols over the local network.

3.3 Software Interface Requirements

EXTINTF_SRS_005: The pi pico variant shall utilize the following:

- Default rp2040 micro-python environment with BLE and WiFi modules.
- Standard BLE direct communication protocols.

EXTINTF_SRS_006: The Pi4 variant shall interface with the following components:

- The Flask framework shall be used to create a web server running on the Raspberry Pi 4.
- Python 3 shall be the primary programming language for backend development.
- Frontend shall utilize HTML5, CSS3, and JavaScript for the web interface development.

3.4 Communication Interface Requirements

EXTINTF_SRS_007 The software shall support HTTP/HTTPS protocols for data transmission. The system does not require email or other direct communication features at this stage. All user interactions shall be managed via the web interface.

4. NON FUNCTIONAL REQUIREMENTS

4.1 Security

NONFUNC_SRS_001: The system shall operate in a local network environment and shall not store personal data. All communications between the web interface and the Raspberry Pi shall be conducted over a secure HTTPS connection if exposed to the public internet.

NONFUNC_SRS_002: In direct connection (BLE) mode, the system shall use standard BLE encryption to protect data during transfer from user app, to display device.

4.2 Capacity

NONFUNC_SRS_003: The software shall be capable of handling individual text inputs up to 10,000 characters. Storage requirements will scale according to the number of images and/or text blocks a user elects to store locally on the display device.

4.3 Compatibility

NONFUNC_SRS_004: Minimum hardware requirements for Pi 4 variant shall include:

- Raspberry Pi 4 Model B with 2GB RAM minimum.
- Power supply with at least 3.0A capacity.
- MicroSD card with a minimum of 8GB storage for the Operating System and software.
- Hoysond 3.5 inch 480x32 OLED screen or other compatible display

NONFUNC_SRS_005: Minimum hardware requirements for Pi Pico variant shall include:

- Raspberry Pi Pico (or pin compatible rp2040 based mcu) with WiFi and Bluetooth support (e.g. Pi Pico W model).

4.4 Reliability

NONFUNC_SRS_006: The anticipated mtbf of the device is limited by the mtbf of the display panels, in the case of e-ink, it is expected to exceed ~200,000 hours.

4.5 Scalability

NONFUNC_SRS_007: Given the nature of the device, the display shall support only a single direct connection at a time, until the connection is closed by the user, or times out.

4.6 Usability

NONFUNC_SRS_008: The web interface shall be designed to ensure that users with minimal technical expertise can input and submit text and/or images without prior training. The system shall aim for a user-friendly experience with a focus on simplicity and clarity.

4.7 Other

Not Applicable or N/A: There are no additional non-functional requirements specified at this stage of development.

5. QUALIFICATION PROVISIONS

This section identifies the qualification method(s) used to verify each requirement. This information is generally provided in a table, with an example shown below. Definitions of the qualification methods are usually provided before the table.

Sample text:

Qualification in this specification is interpreted as requirement verification. The following are the base definitions for the verification methods.

A – Analysis: Use of analytical data or simulations under defined conditions to show theoretical compliance. Used where testing to realistic conditions cannot be achieved or is not cost-effective. Analysis (including simulation) may be used when such means establish that the appropriate requirement, specification, or derived requirement is met by the proposed solution. Examples include the reduction, interpretation or extrapolation of test data.

D – Demonstration: A qualitative exhibition of functional performance, usually accomplished with no or minimal instrumentation. Demonstration (a set of test activities with stimuli selected by the developer) may be used to show that the CSCI, or a part of the CSCI, response to stimuli is suitable (e.g. observation of fin deployment, etc.). Demonstration may be appropriate when requirements or specifications are given in statistical terms (e.g. mean time to repair, etc.).

I – Inspection: The examination of the CSCI code against applicable documentation to confirm compliance with requirements. Inspection is used to verify properties best determined by examination and observation.

T – Test: An action by which the operability, supportability, or performance capability of the CSCI, or a part of the CSCI, is verified when subjected to controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis.

An example verification table is provided below, but may be replaced by tables auto-generated from the requirements management tool, if used.

Table IV. Requirements Verification

SRS Req. ID	Paragraph Title	Verification Method
FUNC_RPI_001	The firmware shall support e-ink and lcd display panels.	Demonstration
FUNC_RPI_002	The firmware shall support caching of user data.	Test
FUNC_RPI_003	The firmware shall support direct connections from the user applications via BLE.	Test

SRS Req. ID	Paragraph Title	Verification Method
FUNC_RPI_004	The firmware shall support connections from the user applications via local wifi/networks.	Test
FUNC_RPI_005	The firmware shall display user images on the attached e-ink or LCD panel	Demonstration
FUNC_RPI_006	The firmware shall display user text input on the attached e-ink or LCD panel.	Demonstration
FUNC_RPI_007	The firmware shall rotate displayed images/text according to user provided time intervals.	Demonstration
FUNC_COM_001	The user apps shall use standard http/s for connecting to the display over local networking.	Inspection
FUNC_COM_002	The user apps shall use standard BLE for direct device connections over bluetooth.	Inspection
FUNC_COM_003	The user apps shall allow users to browse for and upload images from their local file system.	Test
FUNC_COM_004	The user apps shall allow direct input of text to be sent/displayed on the device.	Test
FUNC_COM_005	The user apps shall allow users to specify a time period for how long the data should be displayed.	Test
FUNC_WEB_001	The web application shall detect unsupported browsers and display an error message.	Test
FUNC_IOS_001	The iOS app will request standard bluetooth and network access on first launch, and perform safety checks for permissions on subsequent launches.	Test
FUNC_IOS_002	The iOS app shall support saving session and connection data between app launches.	Test
FUNC_IOS_003	The iOS app shall support manual configuration of device/connections.	Test
EXTINTF_RS_001	The software shall conform to the specified layout and design guidelines.	Demonstration
EXTINTF_RS_002	The web interface shall support the detailed text input design guidelines specified.	Demonstration
EXTINTF_RS_003	The iOS app shall use native tab-switching controls for swapping between text/image modes.	Demonstration
EXTINTF_RS_004	The firmware for the display device shall run on both the RPi4 and Pi Pico W.	Test
EXTINTF_RS_005	The Pi Pico firmware shall utilize micro-python with wifi and BLE modules.	Inspection

SRS Req. ID	Paragraph Title	Verification Method
EXTINTF_SRS_006	The RPi4 firmware shall utilize python 3 and Flask.	Inspection
EXTINTF_SRS_007	Network communication with the web application shall be made via standard HTTP/HTTPS	Inspection
NONFUNC_SRS_001	Neither the firmware nor user apps shall store or harvest any personal/unique identifiers from users.	Inspection
NONFUNC_SRS_002	Direct connections via BLE shall utilize standard encryption to protect data during transit.	Inspection
NONFUNC_SRS_003	The display devices shall support limited caching up to their individual storage limits.	Test
NONFUNC_SRS_004	The RPi4 firmware shall meet the minimum specs as outlined.	Demonstration
NONFUNC_SRS_005	The Pi Pico firmware shall meet the minimum specs outlined.	Demonstration
NONFUNC_SRS_006	The display hardware used shall meet the specified mtbf ratings.	Inspection
NONFUNC_SRS_007	The device shall support only a single direct connection at a time.	Test
NONFUNC_SRS_008	The web interface shall be accessible to those with minimal technical knowledge/training.	Demonstration

6. NOTES

6.1 Acronyms and Abbreviations

Table VII. Acronyms and Abbreviations

Abbreviation	Full name
BLE	BlueTooth Low Energy
MTBF	Mean Time Before Failure
Pi Pico	Raspberry Pi Pico W
rp2040	alternate abbreviation for Raspberry Pi Pico W
RPi	Raspberry Pi 4 B
WiFi	802.11n (2.4Ghz) wireless network standard