SoftwareProject Management Plan

“Project Thisplay”

October 8, 2023

Java The Hutts

Computer Science Dept CSUN

Professor Dantes

Revisions

Version	Primary Author	Description of Version	Date completed
1.0	Entire group	Initial revision	10/8/2023
1.1			
1.2			
1.3			
1.4			
1.5			

Table of Contents

1. Introduction

1.1 Project Overview

Purpose

To create a simple and affordable, wireless “smart” display along with mobile and web applications for the user to interact with/send data to be shown on the display.

Scope

The display will feature a simple API, accessible over Bluetooth and/or wifi, capable of accepting text and/or color images from the users. Mobile and web applications will be available for users to configure and interact with the display over the wireless API. The display will remain functional so long as sufficiently powered, even when not connected to the companion applications.

Assumptions and constraints:

- For the web application we assume that users are running a Chromium (i.e. Google Chrome compatible) web browser, and that the host machine has access either to local LAN networking, WiFi, and/or Bluetooth.
- For the display to use wifi, the user must have an 802.11n compatible wireless network due to the rp2040 not supporting 802.11ac or newer standards.
- For the iOS application, we assume the user is running iOS 16 or newer.
- For e-paper displays, color support is limited to only 7 color ACeP panels, due to the lack of consumer availability of higher color depth (e.g. Kaleido™, Gallery™ 3, etc.) displays.

1.2 Literature Review

What have you researched on the project? Don't forget footnotes

For example:

- After researching many boards we decided on the Raspberry Pi Pico W and Raspberry Pi 4. It offers wireless connection via LAN and Bluetooth which helps us for our embedded system project which requires connection and the price point is great and the company has great documentation and many guides to help us with our board.[1]

- The display was chosen due to the epaper display having great power consumption as well as epaper having superior visibility which is important for our project. This display also has compatibility to any Raspberry Pi board so it is a good match.[2] [3]
- Swift and SwiftUI was chosen since Apple developed and has a lot of documentation to support it is optimized and uses LLVM compiler tech to improve speed.[4] [5]
- Integrated Wireless Connectivit**y**: The built-in Wi-Fi capabilities of the Raspberry Pi 4 allow seamless communication with various devices, eliminating the need for external wireless modules. As per Upton et al. (2020), this integration boosts the efficiency of IoT projects, reducing potential points of failure. Upton, E., & Halfacree, G. (2020). *Raspberry Pi User Guide*. John Wiley & Son
- Performance and Versatility: With a Quad-core Cortex-A72 ARM v8 CPU and up to 4GB of RAM, the Raspberry Pi 4 offers desktop-level performance. It's capable of handling various tasks simultaneously, from web server operations to direct hardware manipulation (Wang & Salcic, 2019). Wang, Y., & Salcic, Z. (2019). Design and implementation of multi-threaded operating systems on single-core microcontrollers. *Journal of Systems Architecture*, 97, 279-290
- Broad Community Support: The widespread adoption of the Raspberry Pi series means a vast repository of community-driven resources, tools, and libraries. This not only accelerates development but also aids in troubleshooting and optimization (Richardson & Wallace, 2012). Richardson, M., & Wallace, S. (2012). *Getting started with Raspberry Pi*. "O'Reilly Media, Inc."
- Web Technologies in IoT: According to Zeng et al. (2016), web technologies offer an attractive pathway for IoT interface development due to their ubiquity and evolving capabilities. The marriage of web interfaces with IoT devices broadens accessibility and enhances user experience. Zeng, W., Gu, L., & Guo, S. (2016). The web of things: A survey. Journal of Communications, 11(9), 833-839.
- Real-time Web: Technologies like WebSockets provide real-time, bidirectional communication between web clients and servers, essential for responsive IoT systems (Fette & Melnikov, 2011).Fette, I., & Melnikov, A. (2011). The WebSocket protocol. *Internet Engineering Task Force (IETF)*, RFC 6455.

[1] Raspberry Pi, "Buy A Raspberry Pi Pico," Raspberry Pi, accessed October 8, 2023, https://www.raspberrypi.com/products/raspberry-pi-pico/.

[2] "7.3inch ACEP 7-Color E-Paper E-Ink Display Module, 800×480 Pixels, SPI Communication | 7.3inch e-Paper HAT (F)." Accessed October 8, 2023. https://www.waveshare.com/7.3inch-e-paper-hat-f.htm.

[3] "What Is E-Paper Display Technology & How Does It Work? | Ynvisible," accessed October 7, 2023, https://www.ynvisible.com/news-inspiration/what-is-e-paper.

[4] Apple Inc. (n.d.). *Swift - Apple Developer*. https://developer.apple.com/swift/

5 View fundamentals | Apple Developer Documentation. (n.d.). Apple Developer Documentation. https://developer.apple.com/documentation/SwiftUI/View-fundamentals

2. Project Organization

2.1 Roles and Responsibilities

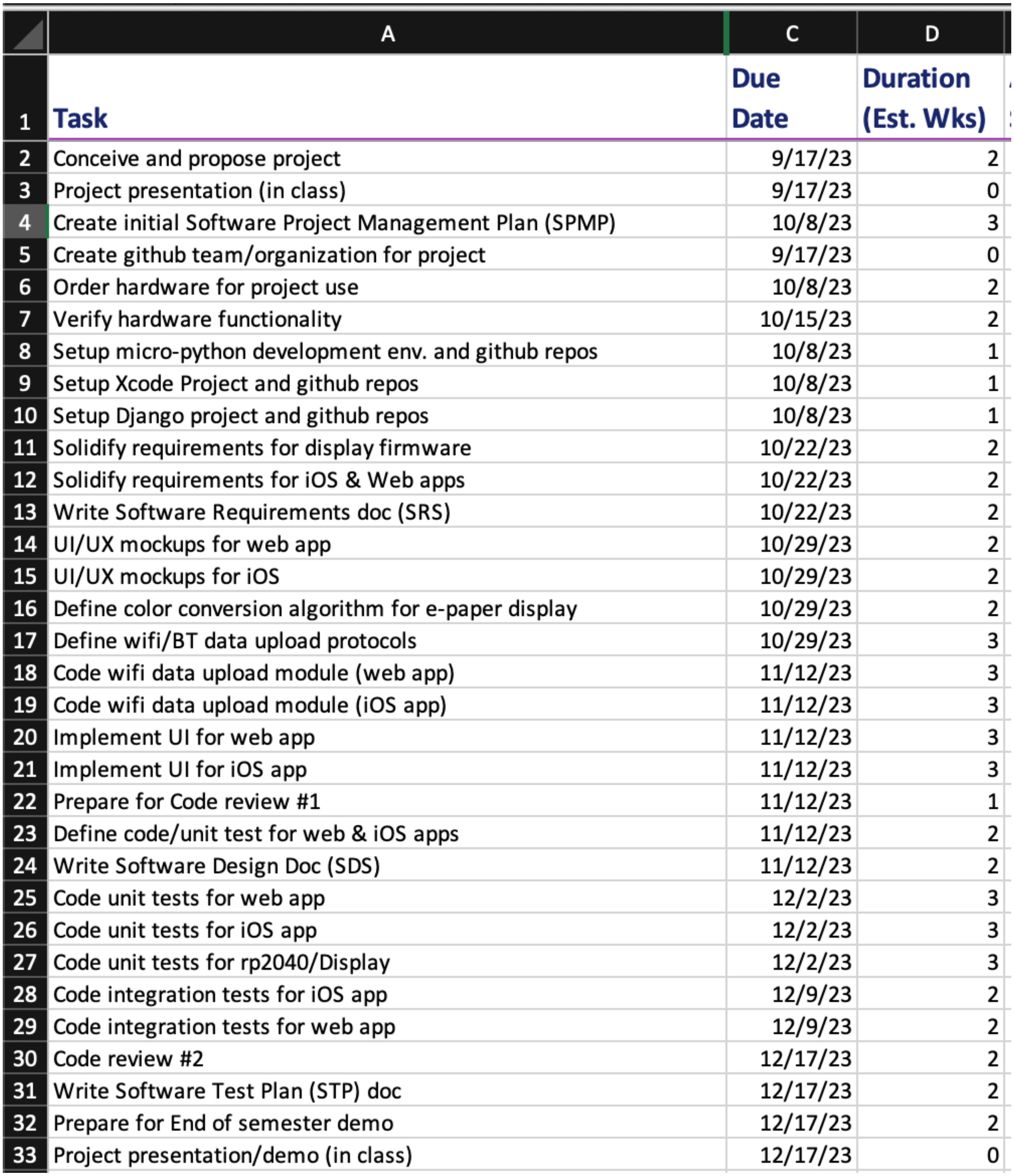
Team Member	Roles	Email
Jocelyn Mallon	Team Lead, iOS/mobile developer	jocelyn.mallon.775@my.csun.edu
Michael DeSantiago	Back end developer/ Front end consultant	michael.desantiago.347@my.csun.edu
Shrey Gahlawat	Front end developer/Tester	shrey.gahlawat.475@my.csun.edu
Erick Barron	Full stack flex developer	erick.barron.785@my.csun.edu

2.2 Tools and Techniques

- Raspberry Pi pico (rp2040) - driver board for display firmware
- Raspberry Pi 4 Model B 4GB - alternate driver board for display firmware
- 7” ACeP, 7-color e-paper display - physical display panel used with the rp2040
- Swift & Swift UI - for native iOS application
- MicroPython - for raspberry pi firmware development
- Django - easy web application development/bootstrapping
- Bluetooth Low Energy - efficient wireless communication protocol for direct connections between display hardware and our web/iOS apps.

3. Project Management Plan

3.1 Tasks

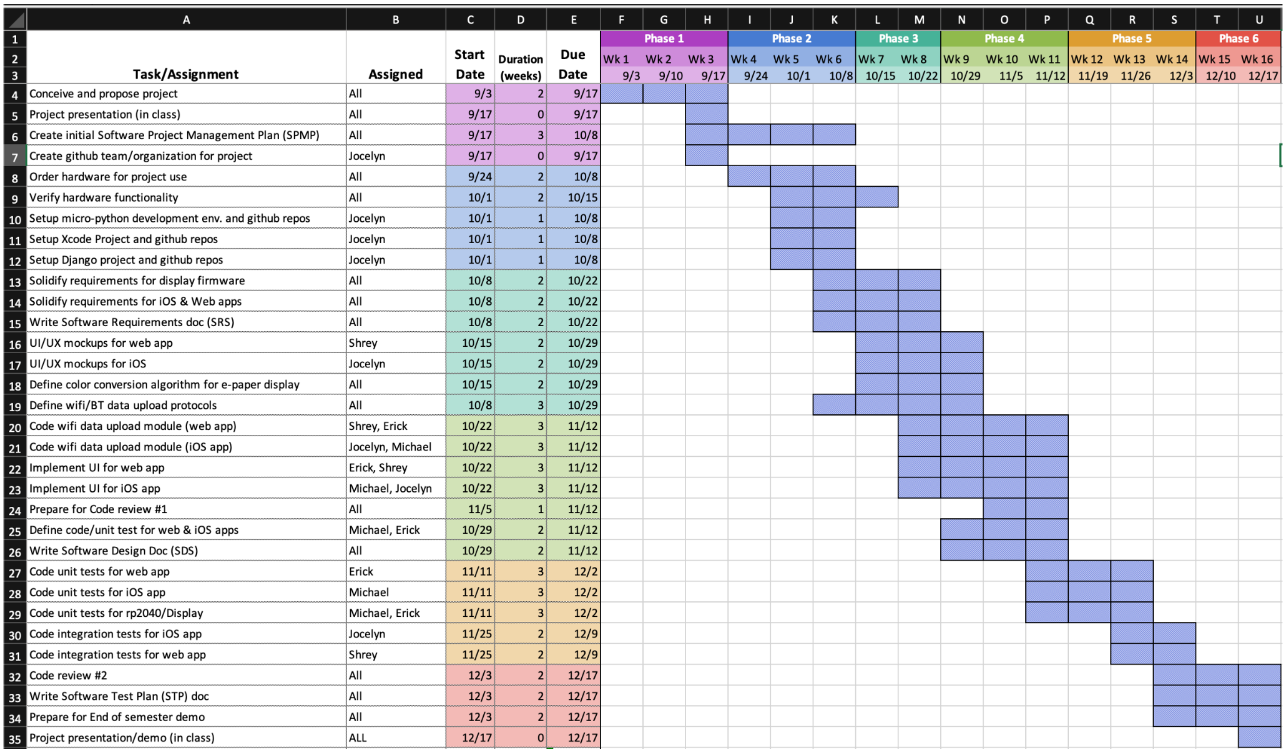


	A	C	D
1	**Task**	**Due Date**	**Duration (Est. Wks)**
2	Conceive and propose project	9/17/23	2
3	Project presentation (in class)	9/17/23	0
4	Create initial Software Project Management Plan (SPMP)	10/8/23	3
5	Create github team/organization for project	9/17/23	0
6	Order hardware for project use	10/8/23	2
7	Verify hardware functionality	10/15/23	2
8	Setup micro-python development env. and github repos	10/8/23	1
9	Setup Xcode Project and github repos	10/8/23	1
10	Setup Django project and github repos	10/8/23	1
11	Solidify requirements for display firmware	10/22/23	2
12	Solidify requirements for iOS & Web apps	10/22/23	2
13	Write Software Requirements doc (SRS)	10/22/23	2
14	UI/UX mockups for web app	10/29/23	2
15	UI/UX mockups for iOS	10/29/23	2
16	Define color conversion algorithm for e-paper display	10/29/23	2
17	Define wifi/BT data upload protocols	10/29/23	3
18	Code wifi data upload module (web app)	11/12/23	3
19	Code wifi data upload module (iOS app)	11/12/23	3
20	Implement UI for web app	11/12/23	3
21	Implement UI for iOS app	11/12/23	3
22	Prepare for Code review #1	11/12/23	1
23	Define code/unit test for web & iOS apps	11/12/23	2
24	Write Software Design Doc (SDS)	11/12/23	2
25	Code unit tests for web app	12/2/23	3
26	Code unit tests for iOS app	12/2/23	3
27	Code unit tests for rp2040/Display	12/2/23	3
28	Code integration tests for iOS app	12/9/23	2
29	Code integration tests for web app	12/9/23	2
30	Code review #2	12/17/23	2
31	Write Software Test Plan (STP) doc	12/17/23	2
32	Prepare for End of semester demo	12/17/23	2
33	Project presentation/demo (in class)	12/17/23	0

3.2 Assignments

Week#	Deliverables/Progress
9/10/23	Progress Report #1
9/17/23	Project Proposal Presentation
9/24/23	Progress Report #2
10/8/23	Software Project Management Plan (SPMP)
10/8/23	Progress Report #3
10/23/23	Software Requirement Specs (SRC)
10/22/23	Progress Report #4
11/05/23	Progress Report #5
11/08/23	Code Review #1
11/12/23	Software Design Document (SDD)
11/19/23	Progress Report #6
12/3/23	Progress Report #7
12/13/23	Code Review #2
12/17/23	Software Test Plan (STP)
12/17/23	Progress Report #8
12/18/23	Project Presentation
12/18/23	Project Source Code

3.3 Timetable



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1						Phase 1			Phase 2			Phase 3		Phase 4			Phase 5			Phase 6	
2			Start	Duration	Due	Wk 1	Wk 2	Wk 3	Wk 4	Wk 5	Wk 6	Wk 7	Wk 8	Wk 9	Wk 10	Wk 11	Wk 12	Wk 13	Wk 14	Wk 15	Wk 16
3	Task/Assignment	Assigned	Date	(weeks)	Date	9/3	9/10	9/17	9/24	10/1	10/8	10/15	10/22	10/29	11/5	11/12	11/19	11/26	12/3	12/10	12/17
4	Conceive and propose project	All	9/3	2	9/17	▨	▨	▨													
5	Project presentation (in class)	All	9/17	0	9/17			▨													
6	Create initial Software Project Management Plan (SPMP)	All	9/17	3	10/8			▨	▨	▨	▨										
7	Create github team/organization for project	Jocelyn	9/17	0	9/17			▨													
8	Order hardware for project use	All	9/24	2	10/8				▨	▨	▨										
9	Verify hardware functionality	All	10/1	2	10/15					▨	▨	▨									
10	Setup micro-python development env. and github repos	Jocelyn	10/1	1	10/8					▨	▨										
11	Setup Xcode Project and github repos	Jocelyn	10/1	1	10/8					▨	▨										
12	Setup Django project and github repos	Jocelyn	10/1	1	10/8					▨	▨										
13	Solidify requirements for display firmware	All	10/8	2	10/22						▨	▨	▨								
14	Solidify requirements for iOS & Web apps	All	10/8	2	10/22						▨	▨	▨								
15	Write Software Requirements doc (SRS)	All	10/8	2	10/22						▨	▨	▨								
16	UI/UX mockups for web app	Shrey	10/15	2	10/29							▨	▨	▨							
17	UI/UX mockups for iOS	Jocelyn	10/15	2	10/29							▨	▨	▨							
18	Define color conversion algorithm for e-paper display	All	10/15	2	10/29							▨	▨	▨							
19	Define wifi/BT data upload protocols	All	10/8	3	10/29						▨	▨	▨	▨							
20	Code wifi data upload module (web app)	Shrey, Erick	10/22	3	11/12								▨	▨	▨	▨					
21	Code wifi data upload module (iOS app)	Jocelyn, Michael	10/22	3	11/12								▨	▨	▨	▨					
22	Implement UI for web app	Erick, Shrey	10/22	3	11/12								▨	▨	▨	▨					
23	Implement UI for iOS app	Michael, Jocelyn	10/22	3	11/12								▨	▨	▨	▨					
24	Prepare for Code review #1	All	11/5	1	11/12										▨	▨					
25	Define code/unit test for web & iOS apps	Michael, Erick	10/29	2	11/12									▨	▨	▨					
26	Write Software Design Doc (SDS)	All	10/29	2	11/12									▨	▨	▨					
27	Code unit tests for web app	Erick	11/11	3	12/2											▨	▨	▨			
28	Code unit tests for iOS app	Michael	11/11	3	12/2											▨	▨	▨			
29	Code unit tests for rp2040/Display	Michael, Erick	11/11	3	12/2											▨	▨	▨			
30	Code integration tests for iOS app	Jocelyn	11/25	2	12/9													▨	▨		
31	Code integration tests for web app	Shrey	11/25	2	12/9													▨	▨		
32	Code review #2	All	12/3	2	12/17														▨	▨	▨
33	Write Software Test Plan (STP) doc	All	12/3	2	12/17														▨	▨	▨
34	Prepare for End of semester demo	All	12/3	2	12/17														▨	▨	▨
35	Project presentation/demo (in class)	ALL	12/17	0	12/17																▨

Additional Material

Definitions, Acronyms and Abbreviations

Any additional information that may be of use to the audience, like any specific definitions, acronyms and abbr.

Example

- RP2040 - short hand for the raspberry pi pico
- RPI - short hand for the raspberry pi 4
- CD - Continuous Deployment
- CI - Continuous Integration
- IDE - Integrated Development Environment
- JS - Javascript
- JSON - Javascript Object Notation
- BT - Short for BlueTooth
- BLE - short for BlueTooth Low Energy

Appendices

Rev 1.0: N/A.

References

Rev 1.0: N/A