

CS CAPSTONE REQUIREMENTS DOCUMENT

APRIL 14, 2020

DUAL-SCREEN APPLICATION

PREPARED FOR

INTEL

MIKE PREMI

PREPARED BY

GROUP 66A

DUO TECH

SACHIN SAKTHIVEL

ROSALINDA GARCIA

DERK KIEFT

MATTHEW FERCHLAND

Abstract

The project's purpose is to solve the issue of having limited applications that are compatible with the Asus ZenBook Pro Duo. To combat this issue, we plan to design and develop a supporting shell application that will allow users to load and save custom window handler configurations. In addition, supporting applications will be built using the shell API to increase the number of applications for the ZenBook. We will utilize tools from traditional applications, in conjunction with new functionalities, to accommodate the additional companion screen on the ZenBook. This document will describe the requirements that must be met in order to validate

the development of the ZenBook application. These requirements include developing a prototype and producing a fully-functional application based on the pre-existing prototype.

CONTENTS

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Definitions, acronyms and abbreviations	2
1.4	References	2
1.5	Overview	2
2	Overall Description	2
2.1	Product Perspective	2
2.2	Product Functions	3
2.3	User Characteristics	3
2.4	Constraints	3
2.5	Assumptions and Dependencies	3
2.6	Apportioning of Requirements	3
3	Specific Requirements	4
3.1	Interfaces	4
3.2	Functions	4
3.3	Performance Requirements	4
3.4	Usability Requirements	5
3.5	Software System Attributes	5
3.5.1	Reliability	5
3.5.2	Availability	5
3.5.3	Security	5
3.5.4	Maintainability	6
3.5.5	Portability	6
4	Signatures	7

LIST OF FIGURES

1	Tentative Gantt Chart for the year.	6
---	---	---

1 INTRODUCTION

1.1 Purpose

The main goal of the project is to improve the usability and functionality of the applications present within the Asus ZenBook Pro Duo. This document exists both for the development of the project and to provide a detailed explanation of the technical requirements for any stakeholders or sponsors.

1.2 Scope

Due to the limited applications that fully utilize the companion screen on the ZenBook, it was proposed to create an overarching shell application that allows users to customize window layouts based on personal preference. The shell application will allow the detection and storage of currently open Windows applications and allow users to reload a suite of windows in one click. After developing the prototype, the group will then convert the prototype into a functional C# application.

1.3 Definitions, acronyms and abbreviations

- **Companion Screen:** 4K 32:9 IPS "ScreenPad Plus" screen directly above the keyboard
- **Main Screen:** 4K 15-inch 16:9 OLED touch display screen on the face of the notebook
- **C#:** A programming language developed by Microsoft that specializes in UI development
- **Notebook:** The Asus ZenBook Pro Duo laptop
- **OS:** Operating System
- **GPU:** Graphics Processing Unit
- **CPU:** Central Processing Unit

1.4 References

N/A

1.5 Overview

Section two of this document will give a more holistic overview of the various prototypes and the developmental specifications of the notebook. This will provide a more informal synopsis of the project to provide more background for a non-technical reader. In contrast, section three will be intended for a more technical audience. It will contain information about the various interfaces, functionalities and software system attributes.

2 OVERALL DESCRIPTION

2.1 Product Perspective

This product originates from the need to develop more software that utilize the capabilities of dual and companion screen systems to enhance the user experience on these new types of systems. While multi-monitor technology has been available for quite some time, the functionality has always been fairly limited. This product will upend the normal use of the second monitor and give the user something truly unique in using the advanced hardware capabilities. The product will rely heavily on the companion monitor technology and focus on the functionality between two touch screen monitors.

2.2 Product Functions

The end product functions to provide a new user experience for dual screen devices such as the Asus ZenBook Pro Duo (or similar companion screen systems). This will be achieved by creating interactions between multiple displays that allow the user to seamlessly interact on two displays at the same time.

2.3 User Characteristics

Users may range from gamers requiring better interaction with their game, to content creators who need more active workspace on their mobile workstation. The commonality between users of the final product will be those who want to take full advantage of new multi screen technologies for greater efficiency in their daily activities. This product will allow them to increase interactive capabilities of their software to match the potential of their hardware.

2.4 Constraints

There are several constraints involved with making applications for the ZenBook. Due to the limited size of the companion screen, application resolution should be properly adjusted based on the secondary display. In other words, users should not be able to notice unattended stretching or blurriness on either screens. In addition to having a small screen, applications should have a maximum of three windows at a time and users shouldn't have to deal with focusing various windows.

2.5 Assumptions and Dependencies

There are several important assumptions that the developer needs to make in order to handle new applications on the laptop. The current specifications for the ZenBook are:

- **CPU:** Intel Core i9-9980HK
- **GPU:** NVIDIA GeForce RTX 2060
- **Memory:** 16/32GB 2666Mhz DDR4
- **OS:** Windows 10

Based on these specs, it would be very difficult to create an application that wouldn't run on the notebook but it is still important to optimize and minimize the CPU, RAM and GPU throughput. These applications also depend on the availability of the ZenBook. To complete this project, we need to have full access to all the capabilities of the laptop and be able to test our product directly on the notebook.

2.6 Apportioning of Requirements

The major portion of this project will involve the development of the basic prototype into a working application. This stage will only begin after designing and developing the shell prototype. Initial prototype development will be pursued by the team members who will flesh out different opportunities to utilize the technology.

3 SPECIFIC REQUIREMENTS

3.1 Interfaces

The interfaces involved in the prototype for this project will primarily be the same across all secondary applications. The general interfaces that will be involved are as follows:

- **Companion Screen:** This is the secondary screen on the ZenBook Pro Duo which is placed above the keyboard. Utilizing this interface and providing a unique experience is the main priority of the project. This will take input from the touch screen and will also react to user interactions with the keyboard and mouse. This will be a key interface for communication with the user and will display output for the user when appropriate. The functionality of the screen (i.e. the precision of the mouse) will be predefined by the laptop or other device. The formatting of the output of the screen (i.e. size ratios) will be customized by the team.
- **Primary Screen:** This is the bigger screen of the ZenBook Pro Duo and is placed as a regular laptop screen. This will not necessarily be the primary interface of the project, as the shell application will be centered around the companion screen. However, this may be important for relaying information to the user. The functionality and formatting of information on this screen will be predefined.
- **Keyboard/Trackpad:** This will be an important interface for receiving user input. For example, the user can type and navigate on both the primary and companion screens. The functionality of the keyboard and mouse will be predefined by the system settings.

3.2 Functions

Due to the nature of this project, once the main shell application is built, secondary applications will be developed. Below are the functional requirements of each application (Update: won't be adding these before the code freeze):

- **Task Management Prototype (Secondary):** Ability for users to create, modify and delete calendar tasks without the overhead of navigation.
- **Adobe Companion Screen integration (Secondary):** Use existing Adobe APIs to customize common use cases in Adobe Ps, Pr, and Lr and implement those in such a way that it's easy to have a one-click companion screen experience.
- **GenderMag Recorder's Assistant (Secondary):** Allow users to record answers to questions in the GenderMag method, take screenshots of appropriate actions, and download completed GenderMag forms, all without obstructing the view or use of the main screen.
- **Shell Configuration Application (Main):** Ability to save and load custom configurations that designates the layout of various window handlers.

3.3 Performance Requirements

The performance requirements for this project are limited. This is because the final result will be based on a prototype. Optimizing the performance of the prototype will not be one of the main focuses. However, the basic requirements for performance are as follows: Minimal Loading Times: The prototypes will be functional and therefore might include parts that have to be loaded. The team should strive to have load times be not more than 4 seconds when applicable. This is to ensure efficiency of use and demonstration of functionality. This will also help ensure that users are able to

use the program with minimal frustration. **Maintain Existing Performance:** In some cases, the prototypes will be made based off of existing open source projects and products. In this case, the prototype developed should try to maintain or exceed existing performance.

3.4 Usability Requirements

The usability requirements of this project are simple as the resulting product will be based on a prototype. However, both the prototype and main application should allow for a good user experience. The requirements for all programs developed in this project are as follows:

- **Ease of Use/Accessibility:** Using the touch screen on the ZenBook Pro Duo should be easy with appropriately sized buttons. The team should strive for appropriate color schemes (e.g. enough contrast, accommodation for colorblind users, etc).
- **Intuitiveness:** The user interface should be simple and understandable. It should use clear and concise language and graphics. All icons and symbols used should be consistent and wherever possible, standardized (e.g. magnifying glass to indicate a search field).
- **Efficiency:** The user should be able to navigate the interface with ease. There should be a reasonable number of steps to accomplish tasks within the program. For example, extra menus, tabs, and windows should have the ability to be minimized. Increasing the intuitiveness and understandability will increase the efficiency of the product.
- **Clarity:** All text used in the prototype should be clear and understandable. The target audience should be able to easily understand the wording. Instructions should be clear, concise, and organized.

These requirements will be evaluated with user testing once the designs are ready. The team will receive user feedback for the prototypes and use the responses to improve the user experience. With this feedback, the prototypes will be able to meet these requirements.

3.5 Software System Attributes

3.5.1 Reliability

While we are only making one functional application, it is important that the prototype runs without errors, bugs, or glitches. This will allow the client to see the capabilities of what we built and get a good grasp on how our program will run without issues.

3.5.2 Availability

We are writing our prototype in order to showcase dual monitor systems, however, we will be writing the program to prioritize the companion screen. We need to also handle the situation where the user only has one screen. We are designing specifically for Windows 10 machines, but if we ever make a production version of the prototype we will need to port it for other operating systems as well.

3.5.3 Security

It is possible that the prototype may use user data (e.g. contacts, email address, etc.). This information should not be used unless necessary. If it is necessary, the team should make efforts to add security measures or note in the documentation

that security measures are needed. Security may not be the primary focus of this project, but it should be taken seriously and considered when developing the design.

3.5.4 Maintainability

All documentation should be kept updated and code should be commented. The prototype may be continued by developers outside of the team after the project time period is over (i.e. if the project is part of an open source product). This means that the documentation and code needs to be maintainable by someone outside of the team.

3.5.5 Portability

Due to the project's scope, it will be difficult to port the functional application to other operating systems. This is primarily because we will need to figure out how to customize the screen view settings based on the user's screen setup as well as their custom preferences.

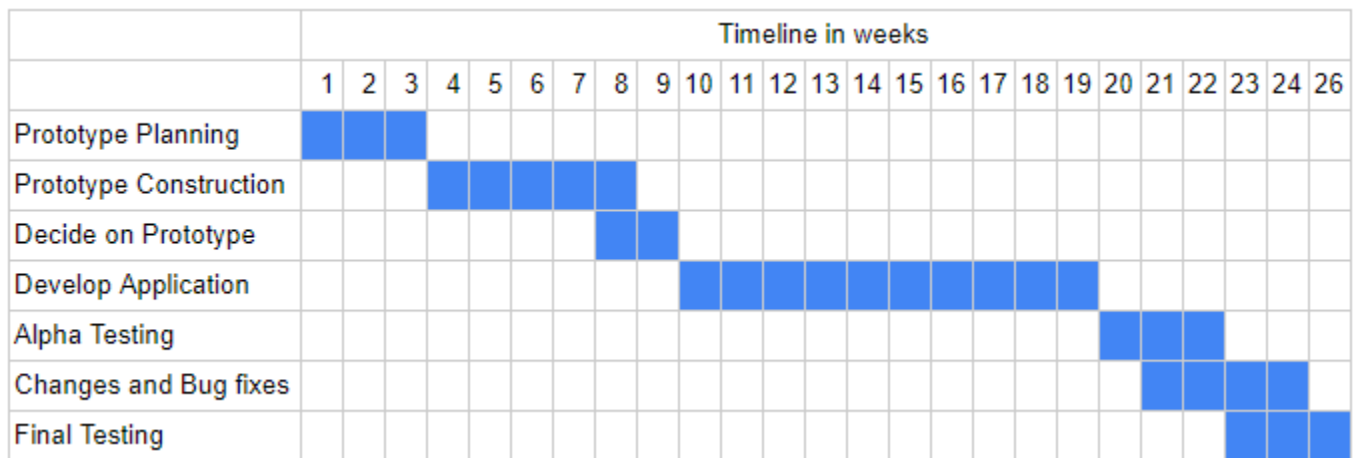


Figure 1: Tentative Gantt Chart for the year.

4 SIGNATURES

Mike Premi

*Signature**Date*

Sachin Sakthivel

*Signature**Date*

Rosalinda Garcia

*Signature**Date*

Derk Kieft

*Signature**Date*

Matthew Ferchland

*Signature**Date*