492 Project Report

Discovery Kiosk

Dakota Rice-Narusch, Cole Chittim, Jesse Ericksen, Megan Hong

10/1/19

Table of Contents

Project Impetus

Currently, Western Washington University's Computer Science Department has a slideshow display at the forefront of the department. It is suboptimal for multiple reasons; most notably, its lack of flexibility for content as well as the difficulty for faculty to upload and manage the files.

Our project aims to replace this slideshow with a touch screen display which, powered by our software, will allow students, faculty, and visitors to further explore the department by interacting with modules. These modules will bring practical content to the fingertips of users and allow for the addition of future modules so that the system can be expanded in the future. By redesigning the current system, we can also provide designated faculty with a better experience when uploading and managing content on the display.

Deliverables

Deliverable ID and/or name	**Description**
Kiosk Application	The kiosk application is an application that displays content dictated by the administrator and provides a touch based interface for users to easily navigate and find information. The application is designed with a modular backbone such that all end user features are to be independent for easy expansion after delivery.
Administrator Portal	The administrator portal is an application which allows the administrator of the kiosk system to upload and manage content on their kiosk, as well as change settings on the device itself.
Slideshow Module	The slideshow module can be enabled on the kiosk to display a slideshow of images that users can view and interact with.
Video Module	The video module can be enabled on the kiosk to enable a media player to showcase videos that users can interact dynamically with.
Map Module	The map module can be enabled on the kiosk to include an interactive map to show where faculty and rooms are located.
Schedules Module	The schedules module can be enabled on the kiosk to the availability of different faculty as well as labs available in the communications building.

Use Cases

UC-1

Use Case:	Logging in to Admin Portal
Goal:	A Kiosk Administrator wants to login to the administrator portal.
Actor(s):	Authorized Administrator, Admin Portal, WWU Authentication Platform
Precondition:	The Administrator has internet access and has access to the Admin Portal website.
Postcondition:	The Administrator is successfully logged in and has access to the Admin Portal.
Primary Course:	1. Admin opens the admin portal website in their web-browser. 2. *Admin Portal* requests authentication using WWU's secure channel. 3. Admin submits their username and password. 4. *Admin Portal* directs request through WWU's authentication platform. 5. WWU successfully authenticates administrator's log-in information. 7. *Admin Portal* redirects the Admin to the *Admin Portal page.*
Alternative Course(s):	4a. Authentication was not successful. 4a1. *Admin Portal* informs user of incorrect log-in info. 4a2. Admin resubmits username and password. 4a4. *Admin Portal* requests authentication through WWU's secure channel. 4a5. Log-in information is successfully confirmed. 5a6. Admin is redirected to the *Admin Portal* page.

UC-2

Use Case:	Uploading image(s) through the admin portal.
Goal:	An administrator wishes to add one or more new images to the slideshow module.
Actor(s):	Authorized administrator, Admin Portal, Kiosk System
Precondition:	The administrator is logged in to the admin portal. (See UC-1) The slideshow module is enabled.
Postcondition:	A new image is added to the slideshow.
Primary Course:	1. Administrator selects the slideshow module. 2. *Admin Portal* pulls up slideshow settings. 3. Administrator clicks the "*add new image"* button. 4. *Admin Portal* opens panel to upload or drag and drop a new image.

	5. Administrator drags new image over. 6. Image is uploaded to *Admin Portal*. 7. Administrator saves slideshow. 8. *Admin Portal* sends new file(s) to Kiosk System via network transfer. 9. Kiosk System updates slideshow module with newly received files.
Alternative Course(s):	5a. Administrator wants to add a file(s) via file location on computer. 5a1. Administrator clicks the "*Specify File Path"* button 5a2. *Admin Portal* opens file navigation window. 5a2. Administrator selects the file(s) they wish to add. 5a3. Administrator clicks "*Submit"* 5a4. *Admin Portal* uploads the image. 5a5. Administrator saves slideshow. 5a6. *Admin Portal* sends new file(s) to Kiosk System via network transfer. 5a7. Kiosk System updates slideshow module with newly received files. 7a. Administrator adjusts image's placement in slideshow order. 7a1. *Admin Portal* updates slideshow order. 7a2. Administrator saves slideshow. 7a3. *Admin Portal* sends new order configuration to Kiosk System. 7a4. Kiosk System updates slideshow module with new order configuration.

UC-3

Use Case:	Removing an image(s) from Admin Portal.
Goal:	The Administrator wishes to remove one or more images from the slideshow.
Actor(s):	Authorized administrator, Admin portal, Kiosk System
Precondition:	The administrator is logged in to the Admin Portal and the Slideshow Module is enabled.
Postcondition:	Image(s) are removed from the slideshow on the Kiosk System.
Description:	1. Administrator selects the slideshow module. 2. *Admin Portal* opens slideshow settings. 3. Administrator clicks on an image they wish to delete. 4. *Admin Portal* opens the image in a panel. 5. Administrator clicks option to delete image in the panel. 6. Image is removed from *Admin Portal.* 7. Administrator saves slideshow. 8. *Admin Portal* sends update to Kiosk System over network. 9. Kiosk System retrieves update and removes selected photos from slideshow.
Alternative	3a. Administrator selects multiple images

Course(s):	3a1. Administrator selects multiple images. 3a2. *Admin Portal* highlights multiple images in panel. 3a3. Administrator clicks option to delete all. 3a4. *Admin Portal* removes selected images from portal view. 3a4. Administrator saves slideshow. 3a5. Portal sends update to Kiosk System. 3a6. Kiosk System receives update and removes selected photos from slideshow.

UC-4

Use Case:	Update display settings
Goal:	An authorized user wants to change a local setting on the display
Actor(s):	Authorized user, Kiosk System
Precondition:	Kiosk is powered on and on the main menu.
Postcondition:	The appropriate setting has been changed on the display.
Primary Course:	1. User taps settings icon. 2. Kiosk System opens authentication field, prompting for Kiosk password. 3. User enters Kiosk password using on-screen keyboard. 4. Kiosk System verifies password. 5. Kiosk transitions to settings menu. 6. Kiosk opens a settings menu. 7. User changes a setting's value. 8. Kiosk System updates the value. 9. User hits Submit button. 10. Kiosk System updates setting. 11. Kiosk returns to main menu.
Alternative Course(s):	4a. Authentication fails 4a1. Kiosk displays failed authentication. 4a2. Kiosk prompts for user information again. 4a3. User resubmits login information. 8a. User taps discard changes. 8a1. Menu prompts to verify user does not want to adjust setting. 8a2. Kiosk returns to main menu without altering settings.

UC-5

Use Case:	Accessing different modules
Goal:	A user interacts with Kiosk System and it's modules
Actor(s):	Student/visitor, Kiosk System
Precondition:	The Kiosk is currently displaying the slideshow.
Postcondition:	The Kiosk is displaying a new module's content.
Primary Course:	1. A user taps the Kiosk screen. 2. Kiosk exits slideshow. 3. Kiosk brings up the main menu with other modules. 4. Student/Visitor taps on the module they want to open. 5. Kiosk System runs the requested module. 6. Student/Visitor interacts with module.
Alternative Course(s):	6a. Return to slideshow module after inactivity. 6a1. Kiosk System waits a specified amount of time for a user to touch screen. 6a2. Any user does not touch screen over this time. 6a3. Kiosk System returns to slideshow module.

Success Criteria

Success Criteria Name	Description	Time/Technical constraint	Goal (for stakeholders, and development team)
File Transfer Requires only Drag and Drop (See UC-1)	File transfers to the admin portal should be as simple as dragging and dropping the files which is a vast improvement over the current system.	This feature will require time to test and determine the optimal method of achieving this means of file transfer.	This system of file transfer is minimal, quick, and simple to use for faculty and development team.
Kiosk System must be modular.	The basis for our kiosk system must be abstracted from the content-specific modules--allowing for our software to be used in a variety of purposes	To properly achieve this with no technical issue, it will require time to plan an efficient and modular system.	This approach is vital as it allows for our software to be easily expanded and modified by future development teams. It also allows our system to be usable with multiple

	outside the CS Department.		clients who have varying needs.
Must support memory and hardware requirements for Dell touchscreen AIO	The development team must be conscientious about memory usage, power requirements, and i/o requirements in design to support this requirement.	Technical Constraint affecting implementation of the core Kiosk application. Won't affect admin portal implementation.	See specifications for computer for exact hardware requirements. Use these requirements as strict limits in development of all software running locally on AIO.
Modules must be able to be easily removed, added and modified.	There should be no limit to the expansion of modules aside from the hardware limitations of the touchscreen computer. The modification of modules must be easy, and not affect other utilities of the kiosk system.	To properly realize this module system will require time to plan and test to ensure its versatile while also easy to use.	This approach allows future research teams to easily modify and expand on our software's features for the ever changing needs of our clients. It additionally needs to be simple to use and understand for faculty operating the system.
Administrator portal and on-board settings require authentication. (See UC-6)	Users of the display must not be able to access display settings, application settings, or other OS settings/files outside our application. The transfer of files must only be handled by a specified administrator.	This feature will require time to test and determine the most secure method for authenticating users while remaining balanced with ease of use.	Use of the administrator portal will require proper network authentication. All settings accessible from the display will require local authentication on the display itself and should not be accessible or visible to standard users interacting with our software.
Kiosk software must be functional from an outside network.	File transfer and implementation of the admin portal must be functional with the kiosk software running on a different network than the admin's host.	Proper implementation of this transfer between these two applications will require a secure solution.	Ensure a secure connection between the two applications that does not rely on both applications being on the same network.

Risks and Mitigations

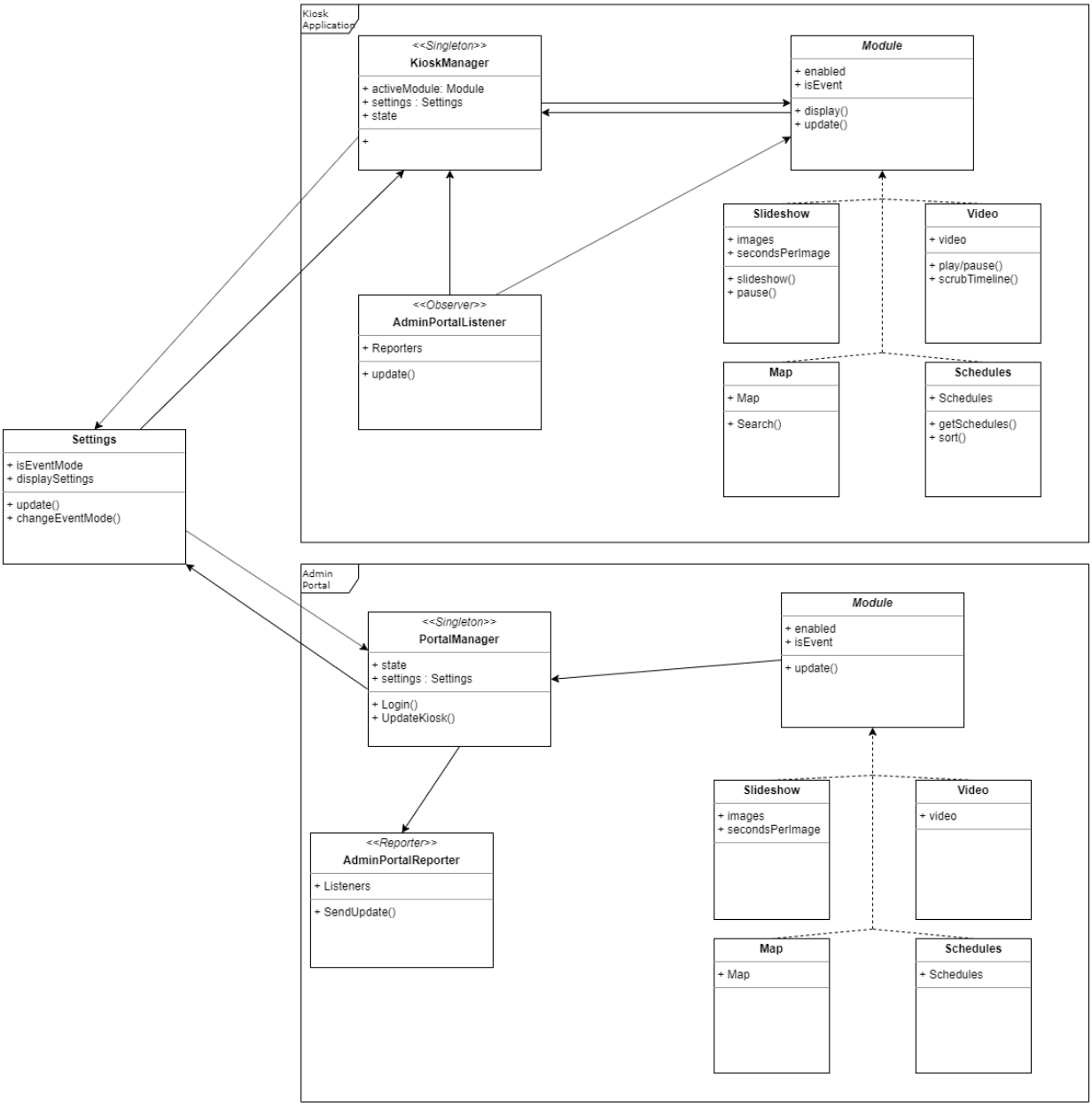
Risks and Mitigation(s)	Description	Analysis	Mitigation/Resolution	Priority
Hacked	The admin portal gets hacked, allowing for others to upload new photos and media.	The likeliness of an individual hacking the administrator portal is highly unlikely.	Utilize authentication standards already in place at WWU rather than design our own authentication platform. Implement two factor authentication as well as system reset.	Low
Kiosk application crashes	An error is thrown causing the kiosk application to crash.	Likely; device upgrades and loss of connection can cause apps to malfunction.	Store error code and timestamp to file in storage, then attempt reboot and restore kiosk to main menu.	Med
Tampering of kiosk system and hardware	The installation of malicious software or hardware via external cables of USBs	A user takes advantage of exposed serial ports to install malware to corrupt the system or hack the system. While unlikely, it would be incredibly easy to attack and easy to prevent.	Require authentication for any local file access and prevent temperament by reducing external access to said components.	Med
Loss of internet connection	The kiosk hardware loses connection to internet.	The kiosk will no longer be able to receive updates from the administrator portal. This is highly likely, but shouldn't be damaging.	System will utilize local storage so as to ensure the system can still operate with current information and will update upon reconnection to the network.	Low
Kiosk application is closed	The device becomes	We must account for the possibility of our	Ensure OS is secured and does not permit	High

revealing access to the OS.	unlocked and users can browse the OS outside of kiosk mode.	program force closing, or a user managing to close our software by some means. In this situation, the OS would be revealed and users might be able to access files and other local resources.	unauthorized changes by anyone other than admins. Upon crashing or unexpected force-closing of application an error code should be saved and stored locally, and device should be rebooted.	

Preliminary Design

Discovery Kiosk is a software system and framework with an end goal of providing the means to easily create and maintain an informational kiosk. Discovery Kiosk will be designed with a modular backbone such that all end user features are to be independent. This means developing a proper framework for this modular system to allow for future implementation beyond the scope of this project. To accompany and showcase this framework we plan to release a core module set with the initial release which will include a slideshow module, a video module, a map module, and a schedule module.

The software system consists of two primary components. The first component being the kiosk application that will run and display content to the end user. This application is designed to showcase content to users as well as provide information while utilizing intuitive touch based controls. The content and information found in this application is curated by the second component, the administrator portal. This administrator portal provides a secure and simple method for faculty to update and curate the content shown and available on the kiosk by modifying what modules are available and how they are currently configured. (The design of these two components can be seen in the figure below)


Kiosk Application
<<Singleton>>
KioskManager
+ activeModule: Module
+ settings : Settings
+ state
+
Module
+ enabled
+ isEvent
+ display()
+ update()
Slideshow
+ images
+ secondsPerImage
+ slideshow()
+ pause()
Video
+ video
+ play/pause()
+ scrubTimeline()
<<Observer>>
AdminPortalListener
+ Reporters
+ update()
Map
+ Map
+ Search()
Schedules
+ Schedules
+ getSchedules()
+ sort()
Settings
+ isEventMode
+ displaySettings
+ update()
+ changeEventMode()
Admin Portal
<<Singleton>>
PortalManager
+ state
+ settings : Settings
+ Login()
+ UpdateKiosk()
Module
+ enabled
+ isEvent
+ update()
Slideshow
+ images
+ secondsPerImage
Video
+ video
<<Reporter>>
AdminPortalReporter
+ Listeners
+ SendUpdate()
Map
+ Map
Schedules
+ Schedules

Deliverable	Task Owner	Task List	Status
Admin Portal <Sprint 1 Deliverable>	Dakota & Megan	Explore environment and framework options. And determine admin portal environment. Find solution for security, prototype.	Not Started
		Develop skeleton according to diagram (see above).	Not Started
	Cole & Dakota	Explore network transfer options.Consider OpenSSL and other alternatives for handling the remote connection. (in relation to administrator portal)	Not Started
Kiosk Application <Sprint 1 Deliverable>	Cole & Jesse	Develop skeleton of kiosk application according to diagram (see above).	Not Started
		Develop model, view, and controller aspects.	Not Started
		Explore network transfer options.Consider OpenSSL and other alternatives for handling the remote connection (In relation to Kiosk Application).	Not Started
Slideshow Module and Video Module <Sprint 2 Deliverable>	Jesse	Task 1	Not Started
		Task 2	Not Started
		Task 3	Not Started
	Dakota	Task 4	Not Started

		Task 5	Not Started
		Task 6	Not Started
		Task 7	Not Started
Map Module and Schedules module <Sprint 3 Deliverable>	Cole	Task 1	Not Started
		Task 2	Not Started
		Task 3	Not Started
	Megan		Not Started
		Task 5	Not Started
		Task 6	Not Started
		Task 7	Not Started

Validation Criteria

Validation ID and/or name	Description	Deliverable(s)	Comments
Slideshow Module	Slideshow displays images and videos, loops through continuously, users can pause, play, and go to the next or previous images and videos.	Slideshow and video module	
Data Transfer Secure	Transfer of data is secure. We test this by making sure huge files don't cause an issue, adding data while interaction with kiosk is taking place does	Admin Portal	

	not cause an error, and adding while kiosk is disconnected does not cause an error.		
Connection Loss	When connection to network is lost the system is still able to operate and receive updates once reconnected.	Admin Portal Kiosk Application	
Module Independence	There exists no co-dependency between modules. All modules should be capable of operating alone on the system as well as all should be capable of running simultaneously on the system.	Admin Portal Kiosk Application	
Touch interaction GUI	The system must implement a touch interface for navigation. Must be able to handle multiple touches at once without system failure.	Kiosk Application	

Sprint 1 Task Assignment

Task Owner	Task ID	Task Description	Comments
Dakota & Megan	Admin Portal Framework	Explore environment/framework options. Determine admin portal environment. Develop skeleton according to diagram (see above).	
Cole & Jesse	Kiosk Application Framework	Develop skeleton of kiosk application according to diagram (see above). Develop both model, view, and controller aspects.	

Dakota & Cole	File Transfer Between Kiosk Application and Admin Portal	Explore network transfer options. Consider OpenSSL and other alternatives for handling the remote connection. Determine best method. Develop prototype.	
ALL	Test Both Applications	Test the Kiosk application and Admin portal and their connection using two devices as well as using the actual dell AIO.	

Sprint 1 Task Completion (Assignment 5)

Task Owner	Task ID	Task Description	Comments

Sprint 2 Task Assignment (Assignment 6)

Task Owner	Task ID	Task Description	Comments

Sprint 2 Task Completion (Assignment 7)

Task Owner	Task ID	Task Description	Comments

Sprint 3 Task Assignment (Assignment 8)

Task Owner	Task ID	Task Description	Comments

Sprint 3 Task Completion (Assignment 9)

Task Owner	Task ID	Task Description	Comments

Final Design (Assignment 10)

<Discuss any changes to the preliminary Design.>

Validation Report (Assignment 10)

Validation ID and/or name	Description	Deliverable(s)	Status
		<Mention which deliverable (by ID and/or name) each validation criteria addresses>	

Conclusion (Assignment 10)

<Discuss the efficacy of the methodology used and the success criteria discussed earlier.>