

Office Door Kiosk

TEAM 28

DR. THOMAS DANIELS

WESTON MORGAN - PRODUCT LEAD

JACQUELINE JOHNSON - PROJECT MANAGER

ERIC RYSAVY - TECHNICAL DOCUMENTATION LEAD

CHRIS DUNCAN - SOFTWARE ARCHITECT

EVAN FOLEY - UI/UX LEAD

PETER LAURION - QA

SDMAY18-28@IASTATE.EDU

SDMAY18-28.SD.ECE.IASTATE.EDU

REVISED: 9/21/2017

Contents

1 Introduction	2
1.1 Project statement	2
1.2 purpose	2
1.3 Goals	2
2 Deliverables	2
3 Design	3
3.1 Previous work/literature	3
3.2 Proposed System Block diagram	3
3.3 Assessment of Proposed methods	3
3.4 Validation	4
4 Project Requirements/Specifications	4
4.1 functional	4
4.2 Non-functional	5
4.3 Standards	5
5 Challenges	5
6 Timeline	5
6.1 First Semester	5
6.2 Second Semester	6
7 Conclusions	7
8 References	7
9 Appendices	8

NOTE: This template is a work in progress. When in doubt, please consult the project plan assignment document and associated grading rubric.

1 Introduction

1.1 PROJECT STATEMENT

Explain what the project is about. What are you trying to do? Solo

Over the course of this project, we will produce custom software (designed for a tablet to be mounted by or on a door) which will be used as an interface for remote communication between professors and students.

1.2 PURPOSE

Going to office hours can be a key part of the college learning experience for many students. However, sometimes life gets in the way. Professors are very busy, and circumstances often arise that may prevent them from attending their stated office hours. What is a student supposed to do in the face of such an issue when they may have time-sensitive questions? The office door kiosk will provide a quick and convenient means of communication between professors and students, even when they're out of their office.

1.3 GOALS

1. Provide professors an interface for remotely relaying information to students
2. Provide students an interactive interface for self service functions during office hours
3. Build an easy to use experience for both user groups
4. Build a professional looking product
5. Practice agile development processes
6. Excellent documentation within the code
7. Automated testing for the product
8. Learn new technologies
9. Get experience taking a project from start to finish

2 Deliverables

These tie in with the goals. What deliverables are necessary to meet the goals outlined in the introduction? Group

1. Admin level client code that updates information for each professor's office hours
2. User level client code for an interactive interface that allows students to perform self service functions

3. Professional interface and easy to use experience
4. Enclosure for tablet
5. An automated testing framework
6. Ample technical documentation
7. Shibboleth Integration

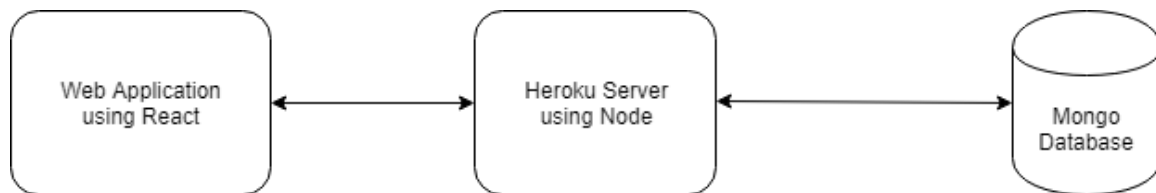
3 Design

Describe any possible methods and/or solutions for approaching the project at hand. You may want to include diagrams such as flowcharts to, block diagrams, or other types to visualize these concepts. Group

3.1 PREVIOUS WORK/LITERATURE

While we have not worked on any similar projects or done research in this topic, we can look at different examples of similar projects. For example, grocery stores use kiosks to provide a self checkout service, which speeds up checkout for most people. We have also experienced kiosks being used at career fairs for filling out forms while waiting in line to speak to representatives. Lastly, we have experience using kiosks for check in queues at places like the doctor's office. All of these examples provide different types of use cases that we can implement into our project. Currently, the downside of these examples is that their specific feature set is pretty limited. Small feature sets lead to smaller user groups. We would like to make sure our project has a large enough feature set, that different people could find many reasons to use the system.

3.2 PROPOSED SYSTEM BLOCK DIAGRAM



3.3 ASSESSMENT OF PROPOSED METHODS

Design Decision: Tablet Kiosk outside of professors door

Alternatives: QR code kiosk

Justification: While a QR code implementation will be cheaper and won't have the implementation problems like charging, it loses a lot of the nice features of having an always on screen for displaying the professor's information and notices. It also adds another step for the student to see the information.

Design Decision: Node.js server

Alternatives: Java server

Justification: Most of our server tasks will be I/O bound and Node.js works faster and scales better than Java. In addition, Node.js is a JavaScript runtime environment for executing JavaScript server side code, so it will be to interface between the two. Also it will make it easier to have developers work on both ends of the application if their written using the same language. Node.js is also growing in popularity and will be a good technology to learn.

Design Decision: React frontend

Alternatives: Angular

Justification: Angular generally has a steeper learning curve than React. Also, our Node server should easily integrate with React, because both are pure JavaScript. React also works really well with mobile, providing more evidence that it is the correct choice. Lastly, React using a combination of JavaScript and HTML markup called JSX, which will easily allow markup to be placed in javascript functions. We feel like this is a very good feature, and plan on using it during development.

Design Decision: No-SQL database (MongoDB)

Alternatives: SQL

Justification: No-SQL databases work really well with Node.js as well as JavaScript because they all natively work with JSON

Design Decision: Web Application

Alternatives: A web application, and a native app for the tablet

Justification: Chrome on android tablets have a kiosk mode so there was no need to have separate app when everything can be put in a single web application.

3.4 VALIDATION

Ideally, we would use two different types of testing to confirm that our solutions work. For simple functionality, we plan on implementing an automated testing framework, that can be run during each new code deployment to make sure that the new code is bug free, and continues to work as expected. However, automated testing can't determine if the problem is solved, only if the code is working correctly. Therefore, we plan to do manual testing for more complicated interactions and checking acceptance criteria. Acceptance criteria is the metric that tests whether the task literally solves the problems we are trying to solve. The acceptance criteria will be specified when creating our tasks. Ultimately, we won't know for sure if the project solves the problem until we test it with different users, including our client, to see if it solves the problems they have. This can only be done after a major prototype has been completed.

4 Project Requirements/Specifications

4.1 FUNCTIONAL

List and explain the functional requirements of the project. This would include all the technical requirements you fulfil during your senior design project. Group

1. Build a secure enclosure for the device
2. Remotely update information on the kiosk
3. Display professor's customizable homepage
4. Leave notes for students
5. Students leave notes for professors
6. Single sign-on with shibboleth
7. Do not disturb mode
8. Scheduler - Way to schedule meetings with professor at kiosk while professor is gone
9. Student check-in queue - Lots of ideas on what this could be exactly but general gist of it would be a way to organize a bunch of students coming to office hours

4.2 NON-FUNCTIONAL

List and explain the non-functional requirements of the project. This is where you would enlist non-technical requirements. This may still be a fundamental deliverable that your client needs at the end of the semester. Group

1. The kiosk must be resistant to thieves and trolls
2. The kiosk must be able to be removed by those authorized.
3. Responsive UI
4. Any number of Professors should be able to have accounts

4.3 ACCEPTANCE TESTS

1. Attempt to remove the device from the enclosure
2. Attempt to remotely update the kiosk from a mobile device
3. Verify that the professor's home page is displayed
4. Verify that notes can be left by the professor and can be read by students at the kiosk
5. Verify that notes can be left by students and received by the professor
6. Verify that signing on with shibboleth signs on to our application
7. Verify that when the kiosk is in do not disturb mode, it cannot be accessed by students
8. Verify that the scheduling works
9. Verify the check-in queue works like a queue and can be seen by both the students and the professor
10. Verify that the UI is responsive
11. Attempt to create a large number of accounts

4.4 TEST PLAN

As there is no current version of the product, there can be no test plan at the moment.

4.5 STANDARDS

All of our standards for writing code were found by searching the web or personal preference. We do not believe that our standard would be deemed unethical by any party. Standards are applicable in our project because we want every team member to look at our code and be able to easily understand it. Having standards on how the code should be formatted allows for someone who didn't write the code to understand it more quickly.

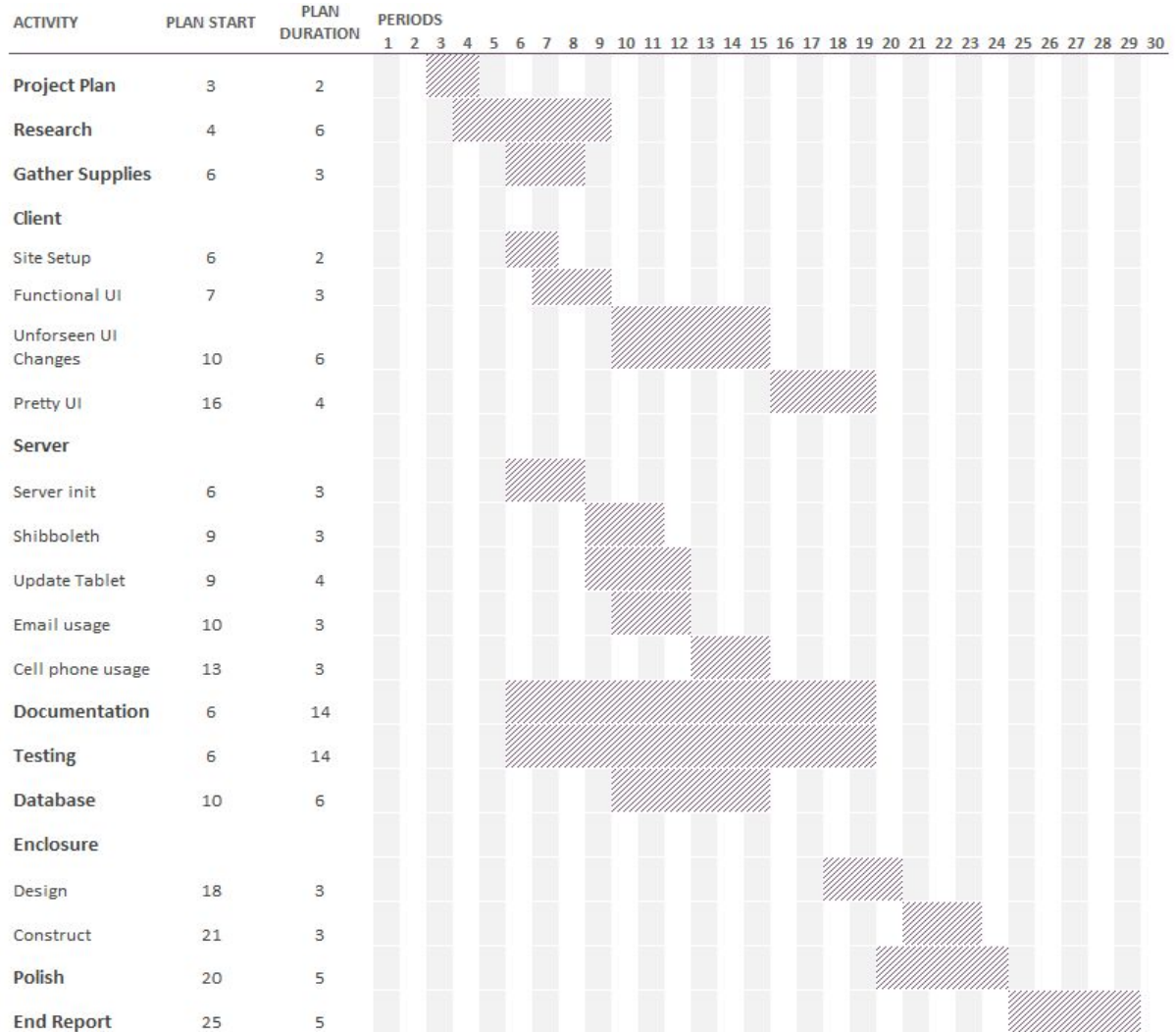
1. Node
 - a. Lines not too long that you can not read them easily.
 - b. Two Spaces for indentations, looks better in web browsers and GitHub.
 - c. Use named functions
2. Html
 - a. Use meaningful title tags
 - b. Use Divs to Divide the layout into major sections
 - c. Don't use Divs excessively
 - d. Use minimal in line css, have unified css file.
 - e. Close your tags
 - f. Use lowercase markup
 - g. Write consistently formatted code
 - h. Avoid excessive comments
3. React.js
 - a. Consistent code organization
 - b. Multi-line JSX, put each element being returned on a separate line
 - c. Conditional JSX, declare empty variable at top, only populate it if the condition is met. Will render either the populated variable or nothing at all.
 - d. 3 or more attributes on a component, display them on multiple lines

5 Challenges

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc. Group

1. Many of us are not familiar with the technologies we plan to use for this project
2. None of us have built an enclosure before
3. Having meetings when we can only meet at one time
4. Keeping the battery of the tablet charged
5. SSO is new territory for us

6 Timeline



6.1 FIRST SEMESTER

The first semester is when the bulk of the project will be completed. By the chart above, the first semester is the first 16 weeks. The site will be up and running with a minimal but functional UI, with server and authentication functionality working. Documentation and testing will be ongoing throughout these changes. Work will be divided according to each team member's predefined role and skills, though slight changes to assigned duties may occur as needed.

6.2 SECOND SEMESTER

The second semester will be centered on polishing and refining the project. Our UI will be adjusted to be both functional and good to look at, and features will be refined and more thoroughly tested. New features may also be added upon client request, time permitting. We tentatively plan on assigning work according our currently assigned roles. Adjustment may be necessary pending how smoothly things run during the first semester tasks.

7 Conclusions

In conclusion, we plan to build a web application using React and Node.js that professors where professors can customize their own webpage to be displayed outside their office doors on a tablet. Our goal is to build a fully featured, clean looking, spiffy version of this idea along with an automated testing framework for it. The requirements we've derived and the schedule we've created will help structure our project so that we can achieve our goals by the end of Spring semester.

8 References

List all the sources you used in understanding your project statement, defining your goals and your system design. This report will help you collect all the useful sources together so you can go back and use them when you need them. likely NA

9 Appendices

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. You may also include your Gantt chart over here.

MAKE SURE THAT THESE SECTIONS OF THE RUBRIC ARE PRESENT IN THE DOCUMENT

Peter-validation and test plan

Jacqueline-Standards and conclusion

Chris-timeline and challenges

Evan-previous work and nature of content

Previous work has been filled out. And I believe our design decision justifications and block diagram take care of the nature of content part. Feel free to verify that.

Weston-proposed design and assessment of proposed solutions

Eric-technical approach and process details