CECS 491A - Section 5/6

December 13th, 2018

Project Plan

[Team Spyderz]

[Product: CheckIt]

Jonathan Asencio [Project Leader] - Back End - 014245983

Alex Philayvanh - Back End - 017508814

Bryan Bare - Back End - 011741260

Kunal Patel - Front End - 013329054

Table of Contents

Introduction

This document explores the details of the CheckIt project. The project will be broken down into eight sprints; three sprints in the Fall 2018 semester, and five sprints in the Spring 2019 semester. Each sprint will consist of tasks that are to be worked on along with its deliverables, if any. Each task will contain features that need to be implemented and each feature will have components to them such as cost, execution, risks. Each sprint will contain a contingency plan along with a deadline. We have provided Gantt charts to visually demonstrate the life cycle of our project. Additionally, a roadmap up to deployment is discussed and even possible future intentions of the project.

Detailed Timeline

The next few pages will show a Gantt chart for our timeline of this project along with written explanations to show what we were thinking. The chart determines when each tasks for our project will be completed and also lays out our future sprint intervals.

[Insert Gantt charts 1 and 2]

Written Timeline

Sprint 1:

Tasks:

- Proposal review
    - Cost:
        - Doing research and coming up with an interesting senior project idea.
    - Execution:
        - A document containing our project proposal and our vision for the application.
- BRD
    - Cost:
        - Learning and understanding features' functional/non-functional requirements and business rules.
        - Making diagrams
    - Execution:
        - A collaborative document with all features, their business rules, and functional/non-functional requirements.
- Technical Specifications Document
    - Cost:
        - Time to research technologies
    - Execution:
        - A document comparing different technologies using diagrams and explanations.

Contingency:

- Given that there were only two tasks, none of the tasks required implementation, and there wasn't any big risks; there aren't any task that we had to get rid of.

Sprint 2:

Tasks:

- Project plan
    - Cost:
        - Time to break down sprints and tasks within
        - Printing paper

    - Execution:
        - A document containing a detailed timeline and roadmap of CheckIt.com.
        - Planning and putting all our tasks onto a gantt chart for this semester and the next.

- High level design document
    - Cost:
        - Time to become familiar with logical flow and network flow examples
        - Printing paper
    - Execution:
        - Whiteboard diagraming brainstorms
        - High level diagrams of actionable sequences
        - Multiple sessions with client to get feedback on designs.
- Revised BRD
    - Cost:
        - Time to go through the BRD and cross check with the requirements list for plagiarism and better explanations
        - Printing paper
    - Execution:
        - A revised BRD with no plagiarism and pass/fail criteria for each feature.

Contingency:

- Due to the importance of the High Level Design Document, we chose to spend a greater amount of time on it than the Project Plan

Sprint 3:

Tasks:

- Authorization:
    - Costs:
        - Time to research, learn, and implement authorization techniques
    - Execution:
        - A well designed authorization system
        - Implementation and testing of system
    - Risk:

        - The tasks take longer than our estimations
- User Management:
    - Costs:
        - Time to research, learn, and implement user management techniques
    - Execution:
        - A well designed user management system (creation, activation, deletion, configuration)
- Password Check:
    - Costs:
        - Time to research, learn, and implement password validation techniques
    - Execution:
        - A well designed password validation algorithm
        - Implementation and testing of various passwords
- Create Database:
    - Costs:
        - Time to research and learn entity framework and integrate it with out system
    - Execution:
        - Creating a SQL relational database using code first method of entity framework

Contingency:

- We will aim to complete all three features for this sprint, however in the case where we run out of time, we will omit the password check implementation

Sprint 4:

Tasks

- Error Handling
    - Costs:
        - Time to research, learn, and implement error handling techniques
    - Execution:
        - Error handling will happen across the entirety of our implementation. Handling planned errors and unplanned errors as they present themselves and
- Search Engine(Web Crawler)
    - Costs:

        - Time to research, learn, and implement web crawling techniques.
        - May need to purchase personal server to host web crawler.
    - Execution:
        - A well designed web crawling algorithm which crawls through listed websites to find prices on items.
        - Adds new websites to crawling list as it finds multiple links to them
        - Efficient runtime and data retrieval
- Privacy
    - Costs:
        - Time to research, learn, and implement privacy handling techniques
    - Execution:
        - A well designed privacy handling system
        - Ensuring there are no data leaks of sensitive information

Contingency:

- In the case where we run out of time, we will lower the complexity of our search feature. Instead of building our own web crawler, we will make use of a search engine.

Sprint 5:

Tasks

- Profiles
    - Costs:
        - Time to research, learn, and implement the creation of a user profile
    - Execution:
        - A well designed profile system with multiple profiles and different abilities with each profile.
- Data Access Layer
    - Costs:
        - Time to research, learn, and implement data access techniques
    - Execution:
        - A well designed data access layer algorithm
        - Efficient runtime data retrieval
- Logging
    - Costs:
        - Time to research, learn, and implement logging techniques

    - Execution:
        - Logging will occur after the implementation of the data access layer.
- Authentication
    - Costs:
        - Time to research, learn, and implement authentication techniques
    - Execution:
        - A well designed authentication algorithm
        - Implementation and testing of system
- Search Engine
    - Costs:
        - Time to research, learn, and implement search engine algorithms
    - Execution:
        - An optimal search engine
        - Implementation and testing of search engine

Contingency:

- In the case where we run out of time. We decided to omit the user profile feature. The core features take priority over the application features.

Sprint 6:

Tasks

- Deals of the Day
    - Costs:
        - Time to research, learn, and implement the design of our feature
    - Execution:
        - An accurate system to determine the best deals of the day using our search engine
        - Deals of the day will check database once per day
- Small Business Promotion
    - Costs:
        - Time to research, learn, and implement the small business promotion design
    - Execution:

        - A platform for small business promotions which allows them to post their products for sale
- Alerts
    - Costs:
        - Time to research, learn, and implement notification and alert techniques
    - Execution:
        - A well designed system to send alerts and notifications via email

Contingency:

- In the case where we run out of time, we will omit the small business promotion feature. Businesses will no longer be able to post their products on our website.

Sprint 7:

Tasks

- Usage Analysis Dashboard
    - Costs:
        - Time to implement dashboard with logging system
        - Time to research and implement UI
    - Execution:
        - Implementation of dashboard with logging system, user access, and UI for dashboard
    - Risk:
        - The tasks take longer than our estimations
- Popularity Analytics
    - Costs:
        - Time to research, learn, and implement analytical techniques
    - Execution:
        - A platform for users to crowdsource on an specific product
    - Risk:
        - The tasks take longer than our estimations
- Log Archiving
    - Costs:
        - Time to learn and design an archiving system
    - Execution:
        - Making design for archiving system on database
        - Implementing archiving system on backend

Contingency:

- In the case where we run out of time, we decided to omit the popularity analytics feature. Users will no longer be able to track product trends and crowd source on a popular item.

Sprint 8:

Tasks

- User Interface
    - Costs:
        - Time to research, learn, and implement User Interface design
    - Execution:
        - A user friendly User Interface working correctly with backend

- User Manual
    - Costs:
        - Time to research and learn how user manuals are structured
    - Execution:
        - A well written and organized user manual for users and developers

Contingency:

- In the case where we run out of time, we decided that we will spend less time designing the User Interface. The user manual will take priority over a well built UI.

Risks

Some risks we face in the implementation of this project include, lack of time, lack of knowledge, or lack of resources. Currently the team consists of four Computer Science seniors graduating in May, given that we are all taking upper-classes to finish our undergraduate degrees, we cannot always devote all our time to this project. Also we are working with new technologies that none of us have worked with before. This can possibly stunt our progress if we cannot implement a certain technology. We are a small team meaning our resources are smaller than other teams, this may affect us when we have more items to complete in a certain sprint. If it comes to the point where we cannot finish all features we will be getting rid of the small business promotions feature.

Roadmap

CheckIt is offering a new way to monitor and buy products. A new wave of ease is being offered through CheckIt's unique ability to be a center hub for searching and monitoring products at their lowest price. Users will be able to search and quickly get the results of the searched product's lowest price points. If the prices still do not meet user needs, they are able to add the item to a watchlist and immediately be notified if a product drops into a desirable threshold.

CheckIt is also tailored to businesses. With CheckIt, businesses are able to promote, post, and propagate their business. Getting rid of extra inventory can be difficult for small businesses who don't necessarily have the ability to reach out of their immediate market area. CheckIt is offering a solution to this by enabling business to directly post their items to our website, broadening their sale potential and getting rid of extra inventory.

Currently CheckIt is in its design phase, with actual application development happening within a month that will lead into the next year. This process will take an estimated 6 months, with an deployment date of May 2019. An addition to this baseline application, we plan to extend our features into a chrome extension and/or a mobile application. Our goal for CheckIt beyond the May 2019 deadline is to have it be a lasting consumer tool and a pivotal business application. Below are the tasks we plan to complete up to deployment.