**Brantly McKendree  
Project Plan  
IT 415 Advanced System Design  
Professor Lazo  
21 March 2021**

**Project Plan**

**Work Breakdown Structure**

The work breakdown structure is the first element of the project plan and is displayed below in Figure 1.1 WBS. It highlights all the main objectives of the Gas Buddy application upgrade project. The project starts with the project design and goes all the way through rolling out a final product. So, the project design is started with a Mach-up drawing. This is going to be a basic drawing of the idea of what the finished product will look like. Next, this design will be pitched to the stakeholders. With any changes being implemented in the next step and repatched again. This will complete the project design phase.

Next, is the project development phase. This phase starts in three parts with the scrum methodology in mind. They are determining server sizes, the beginning of coding for the timestamp front end changes, and the beginning the coding for the server from web scraping code. The scraping will be coded to scrape gas prices from the gas station reporting on google. The timestamp will pull a time from the server for the front end to display when the gas prices are updated. The two development teams will be broken down in two teams on for front end and one for backend/server implementation and there will be a third team working on server hardware and setup. The next section of the project the developers will continue to code and will meet with the scrum master who will gauge the completion of the project and the scrum master will share the results and progression to the stakeholders. In the meantime, Microsoft SQL work will begin and be set up. In the next section the initial server setups will be completed, initial prototypes of the back end and front ends will be completed. The next step is testing of the servers, the timestamps, and the web scraping backend application on the server. The next section is working out any issues and finally rolling out the finalized product after review with the stakeholders.

**Timeline**

The next step in the project plan is determining the timeline. It is depicted in figure 1.2 Gantt Timeline/Breakdown. The steps are as follows with the following dates. Key milestones are highlighted yellow below. The project design should take a little over a week and should be a well-developed design to give stakeholders and development teams a plan. Pitching the ideas to the stakeholders should only take a day, however if changes are needed there is time allotted for this in the design phase. The product development phase will take 6 weeks and a couple days. This will allow for the product development teams to ensure a polished and well-designed product to be produced. Key components including initial server setup, prototypes, and fixing issues that arise are all planned for. Given a week for the prototypes and setups this should be plenty of time to get a working design out with three teams working on each of their sections. If not, testing should be rather quick and is allotted a week as well. If needed the prototypes can bleed into testing time and be tested from there. However, not any more than two-three days as testing is important to ensuring a good finished product. Finally, there are seven days to fix any issues and two days to roll the product out to all consumers via updates.

* Project Design 1.1 – 4/26/2021-5/6/2021
  + Create Mach-ups of final product 1.1.1 – 4/26/2021-4/30/2021
  + Pitch ideas to stakeholders 1.1.2 – 5/1/2021-5/1/2021
  + Implement stakeholder changes 1.1.3 – 5/1/2021-5/3/2021
  + Pitch changes to stakeholders 1.1.4 – 5/4/2021-5/4/2021
  + Refine again if necessary 5/4/2021-5/6/2021
* Product Development 1.2 – 5/7/2021-6/20/2021
* Determine Server Sizes 1.2.1 – 5/7/2021-5/8/2021
* Begin Coding Front End 1.2.1 – 5/7/2021-5/25/2021
* Begin Coding Server Implementation 1.2.1 – 5/7/2021-5/25/2021
* Begin Working on Microsoft SQL Server 1.2.2 – 5/9/2021 – 5/25/2021
* Complete initial Server Setup 1.2.3 – 5/26/2021-6/2/2021
* Create Initial Prototypes (Timestamps) 1.2.3 - 5/26/2021-6/2/2021
* Create Initial Prototype (scraping backend) 1.2.3 - 5/26/2021-6/2/2021
* Test server operation 1.2.4 – 6/3/2021-6/10/2021
* Test Timestamping 1.2.4 – 6/3/2021 – 6/10/2021
* Test Scraping and communication with server 1.2.4 – 6/3/2021-6/10/2021
* Fix all issues from testing 1.2.5 – 6/11/2021-6/17/2021
* Rollout Finished Product 1.2.6 – 6/18/2021-6/20/2021

**Dependencies**

There are a few dependencies that need to be carefully monitored for the project to complete on time. To move on from design to development the stakeholders will need to accept the refined changes so that work doesn’t begin on the project and things are missed or skipped that stakeholders want. The next dependency is going from determining server sizes 1.2.1 to 1.2.2 begin working on Microsoft SQL Server. Without knowing what size storage is needed for the project, the servers can’t begin to be setup. The next dependency is communication between the backend developers and the server setup team so that coding of the communication can be done effectively and proficiently. For testing to begin of each of the separate sections of the project the initial prototypes and setups will need to be complete. Going from 1.2.3 sections to 1.2.4 sections. Finally, fixing all the issues can’t happen until testing is complete and then the same goes for going from fixing issues to rolling out the finished project.

**Use of Tools**

The use of a work breakdown structure, **Figure 1.1 WBS**, gives a guided instruction on the development process and allows for stakeholders to view the progress in terms of what has been completed and what is left to complete when meeting with the scrum master. The WBS also provides a way to distribute the work and provide each team direction. It also provides a visualization allowing for timelines and schedules to made and upheld to reach the final deliverable**. The Gantt Timeline**, **Figure 1.2**, allows for the timeline to be broken down into sections and ensure that each part of the product is completed on time and allows timelines to shift as needed and make the appropriate changes in order to still meet the deadline. These two tools will allow for the final deliverable, Gas Buddy update, to complete on time and in an organized fashion. They will also allow stakeholders to make changes and maintain a chain of communication with the project lead in accordance with timelines and completion dates.

**Figure 1.1 - WBS**

Diagram

Description automatically generated

**Figure 1.2 - Gantt Timeline/Breakdown Gas Buddy**

Graphical user interface

Description automatically generated

**System Design**

**Introduction**

Gas buddy is a gas price analysis application that allows users to determine gas prices from gas stations near them. The current design allows users to enter the gas prices of stations near them that they have verified as well as gas station owners. With this new project the application will automatically update from a server which will scrape the web for gas prices and save them to a database. This database will then communicate with the mobile application and allow users to see gas prices updated by the gas stations on the web. It will also display a timestamp for the users to determine how accurate and when the last update was made to each of the prices. With this update being applied to the application users will still be able to update the gas prices if they are incorrect on the webpages to allow for more accuracy and reliability of the application. The updates from the server should be scanned for hourly if not bi-hourly within the server to ensure updated prices. Thus, making this project a reliability and accuracy update to Gas Buddy increasing the use of the application and increasing consumer downloads.

**Requirements**

For project success a few requirements need to be met. The first of which is scrum sprints daily to ensure work is being completed in a timely manner. This will require the senior developers to determine what has been done and what is left to be done and communicate with the project lead. Also, the stakeholders will need to be available at least thirty minutes to an hour daily to determine if the development process is going as they wish. This can be done by email between the Scrum master and the stakeholders. However, it would be ideal for a once weekly in person meeting if possible, to show progress during the coding phases and testing phases. For the server deployment to be successful the equipment needs to be ready before the team starts their deployment. The team needs to have a plan setup to ensure the server is accurately and efficiently deployed. The server team will need to stay in communication with the backend development team to ensure the communication between the server and application is working properly. The front-end team will need to remain in contact with the server team to ensure their timestamping is lining up accordingly with the server updated times. If all these requirements are met, then the project will deploy successfully and on time.

**Constraints**

There are minimal constraints on this project. The first being time, the project has a deadline and the teams will have to work according to meet this constraint. The server team must complete the server in time for the backend communications to work and the front-end as well. The server will also need to be able to handle the amount of data being saved. The delay of production is the next and this can be mitigated by using the scrum methodology and sprints to ensure that the project is progressing properly. With all teams working in simultaneously they will complete their sections rapidly and potentially complete early so they can begin testing. Finally, changes made during production could cause delays. This is mitigated by following the exact design and going through the first phase of the project which is design with the stakeholders thoroughly to ensure that the constraints of changes not being made mid-way are met. This constraint can also be met by allowing for shifted timelines with the stakeholders.

**Resources**

The Gas Buddy updates are going to require some resources both human resources and technological. The first to discuss is human. There will need to be three teams working in a scrum methodology with the project lead or “Scrum Master.” Of the three teams there will be a team that works on server development which would consist of two to three well trained individuals. It will consist of at least one networking administrator to setup the networking for the server and firewall. The next would be a senior front-end developer and a junior front-end developer for employing the front-end upgrades. They will work on the timestamps and will work with the back-end team to deploy a working update of timestamps from server gas price updates. Another team will consist of a senior backend developer and a junior backend developer. They will work on communication between the application and the server and they will work with both the front-end developers and the server teams to ensure the communications and development process goes well. They will all be meeting with the final human resource which will be the project lead who will lead all the teams through the development process and will meet with teams to ensure the development process completes on time and that the stakeholders are kept involved.

The next resources to discuss are the technological resources. This will consist of five workstation computers which will be used by the development teams and the project lead to remain in contact and allow each to have a place to complete their work as needed. The server team will need a workstation/server machine to create a Microsoft SQL Server for the gas prices to be stored on. This brings in the requirement for not only the Microsoft SQL Server software, but also the networking hardware required. The server will need to be hooked to a network with a firewall to protect information saved within the server.

**System Overview**

The system being developed will involve a server machine that will scrape the internet for gas prices from the gas stations websites across the United States. It will update the prices on the server and give them a timestamp on the application. The application will still allow users to enter if the prices are invalid and enter what they see while out in the field. This will ensure that gas prices are correct and ensure gas is available in emergency situations for consumers. The mobile application will communicate with the server and update gas prices and timestamps according to when they were updated and what the new prices are. This is displayed in figure 1.3 Architecture and Visual application design.

**Documented Detailed Design**

**Figure 1.3 - Architecture and Visual application design**

Graphical user interface

Description automatically generated