Public transportation route planning is always a substantial problem for any metro areas. Even in a small city such as College Station with only one bus system provided by the Texas A&M University Transportation Service, this remains a problem. Currently, the Transportation Service website only provides the daily schedule. To find out how to get from one point to another, one must know where they are currently located on the map to figure out which bus route they need to take. The process becomes even more complicated if they need to take more than one bus to get to the final destination. They have to manually retrieve different bus routes’ schedules to find out what their options are, what the total time is, and how long they have to walk between bus stops or wait at a bus stop between buses. This type of problem interests our group. Via this project, we provide a solution to eliminate this manual process and to also always guarantee to find the optimal results based on the users’ preferences.

Our product consists of three components: a crawler which nightly crawls the bus schedule from Transportation Service website, a route planning engine which takes users’ input from the user interface and combines with prior knowledge about the bus schedules to produce the results, and a web interface for user interactivities. These three components can be independently developed. Therefore, we tasked members of our team based on each person’s skills and interests. We used Python and PyCharm as the programming environment for our back-end tasks, PHP and PHPStorm for our web user interface tasks, and also Symfony2 and Flask to create the web service interface so that the Python back-end can communicate with the PHP front-end. Our product also offers an interactive way for the user to select their locations such a map to click on, the drag and drop ability to re-select, via the means of using Google Map API and Twitter Bootstrap.

In this project, since we only had 6 weeks from prototyping to developing and deploying our product, we decided to choose agile development methodology with scrum model. Our goal is to achieve a process that can support fast-paced development and easy checking on the progress. Each sprint consists of a number of tasks selected as a result of the collaboration between the product owner and the development team. These tasks need to be finished with complete testing so that at the end of each sprint, we can have a workable product with an increasing number of features over time. We decided that each sprint lasted in a period of two weeks with a number of 3 sprints in total.

Through this project, we expect to learn agile development process in action. Getting our hands on the real process with real tasks and real roles among the team members can give us a better picture on what it is and what we can benefit from it. Also, by tasking each member with things we have never done before, we want each to explore a different realm and learn different skill sets from each other. For example, a Python developer gets to learn about web service, a Java developer and a .NET developer get to learn about scripting language and rapid prototyping.

Our goal for this project is to get a workable product satisfying all specified requirements, and a fast-paced development process based on agile methodology with scrum model. Following are the results of our team’s effort in 6 week period and a discussion of them.

Our workable product is a web page which offers an interactive way for users to input their start and end locations, their expected leaving time, and get back the optimal routes based on the earliest arrival time criteria. (see figures in the appendix). The users can click on the map to select the starting and ending points. Also, they can drag and drop the red dots to re-select different points on the map. After searching through all of the possible routes one can take, the search engine will return a result of top 5 routes. Each route can be a combination of different trips in such a way that in each trip the user can take a different means of transportation. For example, to get from A to D, the user has to walk from A to B, then take bus 12 from B to C, then walk to C, and finally take bus 8 from C to D. We limit our results to five possible best routes. For each possible route, we aggregate to a total time it would take, and also display the total waiting time and walking time so that the user can have a better idea of how their options look like.

The users can choose to save their search results to be able to retrieve them at a later point in time. These results are saved as long as the browser is open.

To support that, all of the back-end tasks also needed to be done ahead of time such as importing all geo-data of all bus locations from the data provided by Transportation Service and scheduling the nightly routine to extract the bus schedule from the Transportation Service website. Our web page also has mobile-friendly features with responsive design on three types of device such as PC, tablet, and mobile.

The only thing we did not really achieve in this phase of our project is the performance of the system. We implemented A\* search algorithm with Yen’s approach to find k-shortest paths. However, with a few hundred nodes and thousands of bus times during the day, our algorithm requires a wide range from a split of a second to a over 15 seconds to run, depending on the hardness of the routes involved. The performance can definitely be improved with different techniques which we have already had ideas on how to implement. We saved these items to our product backlog for a later phase.

As for the process, we all agree that scrum model has worked well for a rapid prototyping and developing a product that evolves through time. Our scrum consists of 3 sprints, managed by a Scrum Master. Each cycle in the iterative process involves backlog refinement, sprint planning, developing and testing with daily scrum meeting incorporated in, and finally a sprint review and retrospective meeting.

Daily meetings are to ensure everyone is on the same page before continuing on our development process. It also allows us to quickly and effectively communicate with each other about difficulties. If a change in direction is needed, quick meeting allows this to happen without any delay in the process. This is proved to be an advantage of agile development when our team had to change our take on the algorithm implementation several times to finally come up with a working one with reasonable performance.

Besides daily meeting, pair programming has been a great help in producing high quality code. With pair programming, we tended to consider more design alternatives and arrive at simpler, more maintainable designs. The code produced was more error-prone, in a faster rate. However, it took much more effort than the amount of effort one programmer could spend to solve the problem alone.

To manage our product backlog and sprint iterative development process, we used an online tool called Trello, available at trello.com. This product supported us well in this project with its clean and easy-to-use interface. It allows us to create, edit cards, push them to the correct list, and display the lists in color code in such a way that at any given time, the developers and possibly the managers can obtain the current status of the project in a glance (see figure in appendix). However, one disadvantage this tool has is that it does not provide a way to take a snapshot of the status of all tasks associated to this sprint. We need this feature so that at the end of each cycle, we can have a history of what have been done in the current sprint before starting to move the cards around in preparation for the next sprint.

The sprint review meeting and sprint retrospective meeting being combined into one worked well for us since we did not have a lot of time like in a real software development process. Another thing that worked for us is in this meeting, we were able to give suggested solutions for all of the reasons of failure in the previous sprint so that we can improve in these aspects in the next sprint. This too has been proven to work well.

What we did not do well in this process is to come up with a time estimate for some tasks in which under-estimation cost us the incompletion of them at the end of the sprint. Also, meeting is another time consuming factor in which we did not do a very good job in managing our meeting time with respect to the scope of this project.

In conclusion, through this project we had a chance to practice agile development process, which is a quick and effective way to achieve a workable product where detailed requirements are defined on the go. We have learned how to do things in scrum-fashion. We have learned what makes a sprint failed, and how to correct our mistakes. We have testified that scrum model worked well for small team and also encouraged a very high work ethics. However, we need to do a better job in task estimation and meeting time management. If we were to develop another product that evolves through time, scrum model is the one to follow. As for the technical skill development, each of us has successfully stepped out of our comfort zones and tried to learn new things. We also played different roles in the development process such as developers, testers, and product owner. That allows us to acquire a variety of skills in different fields, adding valuable experiences to our skill sets.