**Required Elements:**

1. A project title and group name including group member names.
2. A project description: a detailed description of the system you plan to develop including development environments (language, platform, and so on).
3. A top level plain text document called README that lists the directory structure of your project directory.
4. A directory structure that includes areas for the project source code, planning documents, meeting minutes, and project reports (each team’s structure may vary slightly)
5. An initial set of meeting minutes (time, attendees, things discussed, and etc.).
6. An initial planning that lays out the big timeline for your project and expected milestones. Include Gantt and/or PERT charts where appropriate. This will be modified in later deliverables. At this point you do not have enough detail to develop a complete plan but you can do a high level planning exercise.
7. Risk management:

**Risk Management:** As a team, you should discuss and identify the top 5 risks for your project. You should develop a plan for (1) monitoring these risks (2) reevaluation of your risks as the semester progresses (3) contingency plans for these risks.

1. A report to your manager (me) that discusses your progress on this deliverable, the team structure you have chosen to use, the names of the team members and their roles, project repository check out and update policies, etc.
2. Member contribution table

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution (%) | Note  (if applicable) |
| Anurag Chitnis |  | 25% |  |
| Gil Wasserman |  | 25% |  |
| Satyanarayana Chivukula |  | 25% |  |
| Nitesh Kumar Sharma |  | 25% |  |

**NOTE**: Submit one document (e.g., deliverableI.doc) that contains items 1, 2, 5, 6, 7, 8, and 9.

Bus Tracking System

**Group Name:** Android Developers

**Group Members:**

* Anurag Chitnis
* Nitesh Kumar Sharma
* Satyanarayana Chivukula
* Gil Wasserman

**Project Description:**

Presently we have 9 buses running around the campus at University of North Texas. Additionally, we have e-ride service which runs at night. Although we have the schedule of all the buses given on the website of transportation at UNT, it is inconvenient to refer that all the time. During the night time when buses are unavailable we have UNT e-ride for our service However, we don’t know where it is in the campus when we need it and how much time it will take to reach us.

Considering these problems in mind, we came up with an idea to develop an android application for students which will display the current location of the buses and e-ride on the google map.

Programming language: Java, Android SDK, Rest API, XML, JSON

Development Environment: Android Studio, GitHub

We have integrated the GitHub repository in android studio so as to ease the process of version control system.

**Minutes of meeting (MOM):**

Minutes of meeting are written record of the meetings of a Team.

The minutes should contain mainly a record of what was done at the meeting, not what was said by the members

Format of our MOM is word document, which we upload in team repository.

These are informal notes that are instantly written record of discussions.

Occurs every Tuesday and Thursday from 10:40 AM to 11:30 AM in the class.

Every team member is required to attend all the minutes of meeting and in case anyone misses the class or MOM, they can login in to the GitHub to remain on the same page in case of any issues or difficulties attending class or MOM.

Every team member is responsible for creating the MOM and updating it on GitHub on every MOM basis.

Some of the discussions of initial MOM are presented below

* Things Discussed
  + 1. Project Requirements, feasibility, scope and technical requirements
    2. Responsibilities, contribution and workload for each team member
    3. Risk Management and contingency plans

Example of our structure/format of MOM.



**Project Schedule:**

We are using Gantt Charts and PERT charts as project control and tracking technique, we use this to track progress of development of our project BUS TRACKING SYSTEM.

Gantt charts show as critical project elements and helps us to understand which activities can be done in parallel.



PERT Charts which stands for Program Evaluation and Review Technique.

Pert charts shows the critical path for the project and any delay in any activity in the path causes a delay in the entire project, hence activities on critical path must be monitored closely.

Both Gantt and PERT chart allow manager to monitor and control project progress and detect deviations.

All the team members are clear about the requirements, scope and feasibility of the project and everyone is on same page.

We have discussed about the individual responsibilities and contribution of each team member to put in to achieve the expected results.

We all are clear about the high level design and coding required for the project. We have almost finalized the software required for the project and just need to test a high level integration of all the required software.

We have decide to use Junit testing for testing of our software product and as of now have decided to do it parallel while developing the code rather doing it at the end of the deliverable.

Hence you can see few of the activities of design, coding, integration and testing are overlapping in Gantt and PERT charts.

Risk Management

As with any software development project, there are risks involved in a variety of areas. First, there is the issue of some team members being inexperienced with the current programming environment. Not all group members have extensive Android experience, or have even had a chance to work with it before. Additionally, the database system used, Firebase, is new to all the group members. As a result, there will be a bit of a learning curve in both areas that may introduce unexpected delays and issues during development. To mitigate this risk, the group will assign the bulk of the early work in Android to the group members with more experience working with it, while leaving more general Java programming to the other developers. As Firebase is new to the entire development team, two members of the team will be involved in the initial setup and will assist the other two group members after they have a better understanding of Firebase. This is opposed to the alternative of relying on just one group member to gain an expertise in the software. Using two group members will restrict the amount of work that they can accomplish in other areas, but guarantees that a major backbone of the app has as much effort and communication involved as can be spared.

Another significant risk with this project is that it requires a working relationship with the UNT Transportation department, as well as as other transportation departments/companies for future development. If a bus driver chooses not to participate and allow the app to track their bus' current location, then the app ceases to hold any functionality for the user. There is no definitive way to avoid this risk, but there are ways to mitigate the possibility of this occuring. For one, the app must be easy to use and unobtrusive. The driver must be able to maintain his normal schedule and routine with as little interaction with the app as possible. This will provide them with fewer reasons for not wanting to use an app that will only help their passengers. Additionally, the app must not be done specifically for one university or area. If there is no compromise with UNT, then an agreement could be reached with other universities instead. By not tailoring to only one university, the potential customer base is not restricted to a small area.

The last risk is a common risk with all software development projects. There is always the risk that the project will not be finished to its desired completion within the semester's timeframe. To accomplish the initial planning, development, testing, and bug resolution, there exists a possibility that certain aspects of the application will have to be abandoned. The focus from the start will be on creating a basic core functionality with a strong focus on the security of the application. As time permits, optional features will be included with the application, where the necessity of the each feature will be decided early on in the project's development.