|  |  |  |
| --- | --- | --- |
| **SEC NO.** | **CONTENTS** | **PAGE NO.** |
| **1** | **Introduction** | **1** |
| **2** | **Process Model** | **1** |
|  | **Which Process Model Was Used By The Team?** | **1** |
|  | **How Well Did The Team Follow The Process Model?** | **1** |
|  | **How Did The Team Deviate From The Process Model?** | **1** |
| **3** | **Schedule** | **1** |
|  | **How Well Did The Team Follow The Schedule?** | **1** |
|  | **How Did The Team Handle Falling Behind Schedule?** | **2** |
| **4** | **Team Organization** | **2** |
|  | **What Type Of Team Organization Was Used?** | **2** |
|  | **How Well Was The Team Organization Followed?** | **2** |
| **5** | **Product** | **2** |
|  | **Product Pros** | **2** |
|  | **Product Cons** | **2** |
|  | **What Requirements Were Not Met?** | **2** |
|  | **Product Design Changes** | **3** |
|  | **Future Improvements** | **3** |
| **6** | **If We Started Over Again From The Beginning....** | **3** |
| **7** | **What Was Learned From The Project?** | **4** |

1. **Introduction**

In this final report document the MAD team evaluates their overall development process and the final Lunch Decider product. This document also describes what the team learned from the project and how the team would do it differently if given another opportunity.

1. **Process Model**
   1. **Which Process Model Was Used By The Team?**

We chose to follow the Agile Model Driven Development (AMDD) process model for the Lunch Decider project. We chose this process model because it allows modeling for a few hours and then coding for few days. Model storming sessions take place for a few minutes. We selected this because initially an architecture is built and then we envision it and come up with good model that works perfectly. This enables you to set a viable technical direction for your project and to provide sufficient information to organize your team around your architecture. Model reviews and code inspections are done wanted. Acceptance testing will be mainly performed in AMDD.

* 1. **How Well Did The Team Follow The Process Model?**

The team followed all the phases of the model like requirements envisioning, architecture envisioning. Using this architecture, the team came up with actual system, planning, iterative modeling and coding and acceptance testing in sequence.

* 1. **How Did The Team Deviate From The Process Model?**

For the most part, the model was followed. Model storming sessions were not always performed. Due to falling behind schedule, design and coding of some increments were performed in parallel and most testing was performed during code implementation.

1. **Schedule**
   1. **How Well Did The Team Follow The Schedule?**

From requirements gathering to creation of user interface design we followed the schedule perfectly. For the task of retrieving restaurant data from web based services we tested various API’s and we spent more time to select the best approach. The game design also took awhile to complete. These two tasks resulted in deviations from the schedule.

* 1. **How Did The Team Handle Falling Behind Schedule?**

Falling behind schedule was managed by working more days than planned and adjusting the task responsibilities between the team members.

1. **Team Organization**
   1. **What Type Of Team Organization Was Used?**

Our team used a democratic decentralized organization. Though we have followed it, we have a team leader who monitored and led the project.

* 1. **How Well Was The Team Organization Followed?**

At each and every meeting, we as a group discussed all subjects. The leader divided the tasks among all the team members. Communication among our team members was good and this resulted in increased productivity. In some cases we divided among ourselves to work on different parts of project and had meetings accordingly.

1. **Product** 
   1. **Product Pros**

The final Lunch Decider software meets all of the product requirements which were within our control. Operation of the application is very intuitive and straightforward. The finished product is very useable.

* 1. **Product Cons**

The operation of the application violates Google's TOS. Also, due to limitations of the available restaurant data web service API's, we were not able to meet all of our initial requirements.

* 1. **What Requirements Were Not Met?**

Restaurant cost data was not able to be obtained via any of the available web-based API's therefore restaurant cost was not included as a selection criteria.

* 1. **Product Design Changes**

The two following significant changes were made to the original requirements and design of the Lunch Decider application.

* + 1. Originally the phone was planned to directly use the Google API to build a restaurant database that would reside on the phone. The user would control the update schedule of the database. Instead of this approach, we decided to perform these functions on a web server. We use a Database Constructor php script with uses Google's Local Search API to build a restaurant database that resides on the server. The Database Constructor is automatically run periodically to keep the database up to date. By implementing a server based database constructor and database, the need for a user initiated database update function was eliminated.
    2. A server based Query Handler backend was implemented which receives the user's selection criteria from the phone and then queries the database for all matching results. The Query Handler then randomly selects a restaurant from the list of matches and returns the selected restaurant and 3 others (filler for the game display) to the phone in the form of a JSON string. The server based Query Handler reduces the amount of phone memory required for the Lunch Decider application and it improves the application's performance.
  1. **Future Improvements**

In the event the team were to continue development of the Lunch Decider application, several improvements would be pursued.

* + 1. The game appearance would be improved to provide more of a 3d look. The game functionality would also be improved to an actual physics based Plinko simulation in place of the existing predefined animations.
    2. Sound effects would be added to the game.
    3. Screen shots would be added to the application's help section.

1. **If We Started Over Again From The Beginning.....**
   1. We would better familiarize ourselves with the development environment, tools, and associated capabilities and limitations before finalizing the requirements, plan, and design.
   2. We would develop using a different process model.
   3. We would select a simpler game interface theme. Plinko proved to be too difficult to implement to our satisfaction within the time constraints of the project.
2. **What Was Learned From The Project?**
   1. We learned that it is extremely difficult to design a software product when you are unfamiliar with the development language and environment and their associated capabilities. Because of a lack of knowledge regarding the development language, and a lack of knowledge regarding the capabilities of the Google Android OS, following the initial design and plan was not realistic.
   2. We learned about team organization and communication and how to work in real-time projects.
   3. We learned new skills in Java, PHP, mySQL, and the Google Android SDK.
   4. We learned about system design, software testing, and technical writing.