|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | | | | |
|  | |  | | | |
| Project: | | University Food System (UFS)  CSE 5325 – Fall 2021  Project Management | | | |
| Module: | | Project Scope & Feasibility | | | |
| Deliverable: | | Scope & Feasibility Document | | | |
| Version: | | | [1.0] | Date: | [09/23/2021] |

Prepared by: Prithiviraj Eswaramooorthy 1001860633

TABLE OF CONTENTS

[1. Introduction and Executive Summary 2](#_Toc81756269)

[2. Objectives 3](#_Toc81756270)

[2.1 BUSINESS Objectives 3](#_Toc81756271)

[2.2 SYSTEM Objectives 4](#_Toc81756272)

[3 Project Feasibility, Risks and Metrics 5](#_Toc81756273)

[3.1 Project Feasibility Concerns 5](#_Toc81756274)

[3.2 Project Risks 5](#_Toc81756275)

[3.3 Project Metrics 6](#_Toc81756276)

[4 Project Scope and Process Model 8](#_Toc81756277)

[4.1 Project Process Model 9](#_Toc81756278)

[4.2 Project Context 10](#_Toc81756279)

[5. Assumptions and Constraints 11](#_Toc81756280)

[5.1 ASSUMPTIONS 11](#_Toc81756281)

[5.2 CONSTRAINTS 11](#_Toc81756282)

[6. Project Tasks, Schedule and Cost 12](#_Toc81756283)

[7. Conclusion and Recommendations 14](#_Toc81756284)

[Appendices 15](#_Toc81756285)

# 1. Introduction and Executive Summary

UTA is one of the tier one research university in Texas with around 50k students enrolled in an academic term. When the campus is operating in full capacity, it is observed that students are spending more time in the billing queue and wait time is more when food is being prepared during the peak dining hours. UTA has approached our company to design and build a website and a corresponding mobile application known as **University Food System** to order food online and provide doorstep delivery and pickup option for their on-campus restaurants and nearby restaurants around the campus. With the implementation of this service, students can save a considerable amount of time. In the current pandemic situation where, social distancing is most recommended, students can also order food to be delivered to their resident hall or apartments. Even if student is ordering for pickup, it reduces the chances of restaurants being crowded. So, investing on this product would be of great value and benefits the students.

Currently there are online food delivery options like door dash, grub hub available in the UTA location. But they do not provide services of on-campus restaurants which are most preferred by UTA students. The objective of the application is to provide online order, food delivery, status tracking of ordered food from restaurants and other additional services. The estimated budget for this application is $341,000 and it is to be delivered in a timespan of 3 months. The application also generates an income as restaurants and the vendors will be charged 10% of their revenue to use our application and additionally through advertisements.

# 2. Objectives

## 2.1 BUSINESS Objectives

The following is the list of business objectives:

**Objective 1**: Login – project will be secure, users must register and login before use.

**Objective 2**: Payments: System should be able to handle payments (Meal plan and Credit cards only)

**Objective 3**: Delivery: System should allow students to track their order and get their food delivered or select an option for pick up.

**Objective 4:** Registration: Registration can be done only by using University id of students.

**Objective 5:** Restaurants:System should allow vendors to add their new restaurants and add or modify their menu items.

**Objective 6:** Orders:System should allow to place the food order and provide options to modify the items in the order or cancel the order within five minutes from the order placement time.

**Objective 7:** Communication: System will send text messages and email to the registered mobile number and student email upon order placement and status updates like food is prepared, ready for pickup, out for delivery etc.

**Objective 8:** Favorites list: The System will allow to add food items to favorites list from the previously ordered list. The item should be ordered at least once from the same restaurant. System will also by default add the food items to the favorites list if they are ordered more than once from the same restaurant.

**Objective 9:** Search: The homepage of the system will have a search bar and we can search for food items or the restaurants in it.

**Objective 10:** Review and Rating: System should allow to rate and review the food items and the service provided on a scale of one to five stars and add a short comment as feedback.

**Objective 11:** Advertising: System should provide advertising space that displays any combo offers, coupons from restaurants. This could be a source of revenue for the app.

**Objective 12:** Location: System shows the map view with our location and suggests the nearest restaurants at that location.

**Objective 13:** Premium Membership: System provides option to get a premium membership by paying a minimum amount for which, there will be no delivery fee and faster order processing in priority in peak hours.

**Objective 14:** Complaints and Refund: System provides an option to mention any fault or complaints on their food item for the one hour after the food delivery. The request goes to the restaurant and the restaurant can provide a refund of amount if it is a fault in their end.

## 2.2 SYSTEM Objectives

The following is the list of system objectives:

**Objective 1**: Both web-based and android application will be provided.

**Objective 2**: Google Search will be integrated into the system for search.

**Objective 3**: The User Interface would be developed using Angular framework which helps to build interactive and dynamic single page applications.

**Objective 4**: Java will be used in the middleware which uses Restful Webservices API calls to interact with the frontend framework and the Database.

**Objective 5:** PL/SQL will be used to add, fetch, modify the data in the Oracle Database both for Students and Restaurants data.

**Objective 6:** Jira – a project management software will be used to create workflow to plan, track and release the developed application.

**Objective 7:** GitHub will be used as version control system to collaborate the team work efficiently and merge all the development work into a single source.

**Objective 8:** Jenkins will be used for Continuous Integration/ Continuous Deployment. It will be used to take build of the latest source from Git and Deploy them in Dev and Test environments.

**Objective 9:** Integrate Push Notifications to send email and SMS alerts about the orders.

**Objective 10:** Integrate Google Maps API to provide suggestions about nearby restaurants based on current location.

**Objective 11:** The minimum hardware requirements to run the required software are:

Laptop/Desktop:

Operating System: Window 8

RAM: 4 GB

Storage: 100GB of free storage

Processor: Intel I3 with 1.2Ghz processor

Mobile: Android 8 or above

# 3 Project Feasibility, Risks and Metrics

Project feasibility and metrics are summarized below:

## 3.1 Project Feasibility Concerns

**Market Readiness:**

Similar food services applications are already available in the market. Grubhub and DoorDash are two applications which are similar to UFS. System provides all those features with an enriched look and additional features like refund option and recommending favorites from our past orders, which are new compared to other services. To make students use and accept this system, System will provide credits for each order placed using the student Id. Also, Referral credits will be provided if a student refers another student. These credits will be used as discounts to the final billing amount.

**Technical Issues:**

The number of students using the product might substantially increase in a large number over a period. The system could face scalability concerns and web security threats to handle increased traffic and to accommodate new features. Multi factor authentication and SSL certificates should be updated to ensure user data is secured.

**Resources:**

Resources to handle the application without crashing when there are more users in peak dining hours would be a concern. If the servers and databases are outsourced, there should be quick response and resolution from the concerned team. Alternatively, we can have dedicated servers and backup servers to utilize if anyone crashes. This in turn will increase the cost of the product.

## 3.2 Project Risks

**Risks:**

**Time** - Delivering the end-to-end tested product within the timeframe of three months could be a risk if the required resources were delayed or developers hired were not available due to unprecedented reasons during the planned duration. In such case there could be lack of time in testing phases like Unit Testing, Integration Testing, Smoke Testing, System Integrated Testing, User Acceptance Testing may deliver the product in an unstable state.

**Database Encryption** - Like all other application, UFS also stores all the user data like student id, passwords etc. Database contents should be confidential and couldn’t be accessed by anyone including Database Administrators. Malware attacks and security threats should be eliminated.

**Cost** - The overall incurred cost of the project can sometimes increase more than the planned budget when some rework of incorrectly developed modules or when the customer changes the requirement. Other situations like outsourcing contractors, unanticipated travel requirements, change in charges for any third-party software licenses and services could also reduce the profit.

**Mitigation:**

**Time** - This can be mitigated by hiring more resources to meet the expected delivery date. Two highly skilled Quality Assurance resources and database administrators are additionally required. Regressively testing the progressing product every two weeks will ensure all functionality are handled and new changes in the requirements from the customer could also be tested as soon as it is developed.

**Database Encryption** - For security reasons the data should be encrypted before storing in the Database. Some measures to keep the data secured is:

* Using random number generator to create salt of 16 bytes or longer.
* Hash stretching using password based key derivation function with hashing algorithms like HMAC-SHA-256.
* Increasing the Number of Iterations in hashing algorithms.

**Cost** - To overcome budget issues, raise the quotation to a maximum of 20% more than the estimated budget to the customer and check if customer is accepting to pay for it. It would be an ideal solution to retain the profit.

## 3.3 Project Metrics

Initially upon launch of the product, simultaneously 1000 students would be using the application as a trial run. When the university is functioning at full capacity, the application usage is rapidly expected to grow up to 10000 students at the same time when all features and functionalities are released.

**Effort Variance:**

The difference between the planned outline effort and the effort required to undertake the task is called Effort Variance.

**Effort Variance = (Actual effort – Planned Effort) / Planned Effort \* 100.**

(If Effort Variance is 0 means the actual required effort is implied, if negative it is less the planned effort. If effort is positive, more than the planned effort is implied)

-50 means only 50% of the planned effort is attained. 50 means 150% of the planned effort is attained.

**Requirement Stability Index:**

Provides visibility to the magnitude and impact of requirements changes.

**RSI = ((Number of Changed + Number of Deleted + Number of Added) / Total Number of Initial requirements) \* 100.**

On a scale [0 -100] (0 means the product does not have any requirement change, 100 means all the requirements are modified)

**Schedule Variance:**

The difference between the scheduled completion of an activity and the actual completion is known as Schedule Variance.

Schedule Variance =

**(Actual Calendar Days - Planned Calendar Days) / Planned Calendar Days \* 100**

On a scale [0 -100] (0 means the product is developed on time, 100 means it took twice the estimated time)

# 4 Project Scope and Process Model

Project scope includes the following:

The major functions that will be included in the first release are:

1. **Registration:** The Students will be able to successfully register with their University Id as login Id
2. **Login:** The Students will be able to successfully login with their credentials
3. **Payments:** The Students can add their meal plan or add their credit cards for payments.
4. **Search:** Search option provides both search of restaurants and food items.
5. **Orders:** Food orders can be placed. Provides options to customize the meal to add or remove toppings, order sides, drinks etc. Within five minutes of order placement, Order can be cancelled, or items can be modified.
6. **Notifications:** Once the order is placed, SMS and Email notifications would be triggered.
7. **Status Tracking:** Once the order is placed, order preparation and delivery status could be tracked in map.
8. **Favorites List:** Favorites tab displays frequently ordered items and allows students to add items that are frequently added.

The following is a list of items out of scope:

1. Post project maintenance
2. Tax issues
3. No Support for IOS Devices
4. Does not support Apple pay, or other payment providers like Gpay, Venmo, Cash App.
5. Advertisement, Sales and Marketing of this application.

## 4.1 Project Process Model

The overall time available to develop the working model of University Food System is three months. There are variety of SDLC process models followed in the industry. For this application, following **Agile** model would be a better approach.

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in sprints. Each sprint typically lasts from about one to three weeks. Every sprint involves cross functional teams working simultaneously on various areas like

* Planning
* Requirement Analysis
* Coding
* Unit Testing
* Acceptance Testing

At the end of each sprint, a working product is displayed to the customer and important stakeholders. The reason for following Agile model here is we can build and release small phases of the application and interact with the customer for feedback or changes required in it. Agile model helps greatly in accommodating the changes in the requirements of the user throughout the development process. Some benefits of using Agile model over others are:

* Higher quality product
* Gives flexibility to respond to market changes.
* Interaction with client frequently gives clear view of what is needed.
* Costs are more easily controlled using Agile.

## 4.2 Project Context

The High-Level Context Diagram inscribing all the functionalities of the UFS is given below:

Diagram

Description automatically generated

**Context Model of University Food System**

# 5. Assumptions and Constraints

## 5.1 ASSUMPTIONS

The following is a list of assumptions:

* Only UTA students can use this product.
* Application is initially free, advertisers will pay for the ad space.
* The application shows restaurants only within 3 km radius around the campus.
* Restaurant will be charged 10% of their revenue made through the app.
* Vendors can cancel the order anytime based on resource availability.
* The food delivery time are subject to traffic conditions, and can differ from the estimated time
* Computers in the restaurants have minimum system requirements to run the application.
* All developers have adequate knowledge and are highly skilled.

## 5.2 CONSTRAINTS

The following is a list of constraints:

* Payments are handled by trusted payment gateways, and if there are any downtime in payments, it couldn’t be handled by development team
* Use of different technologies for any new requirement changes requires training to be provided to the developers.
* Additional 2 highly skilled QA Testers are required to deliver the product on time.
* The overall cost of development should not exceed the estimated cost

# 6. Project Tasks, Schedule and Cost

The project plan and tasks that will be employed in the project are:

* This development plan requires 2 frontend (UI/UX Developers), 2 Backend Developers who will develop both the website and android application simultaneously.
* Two QA Testers will be testing the developed modules and raise any defects found.
* Two Database Administrators will be deployed for tasks like Data handling, Security, performance analyses of database.

**Overview of Product Development Plan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task ID | Task Name | Task Duration | Start Date | End Date | Resources Involved |
| 1 | Sprint Planning | 2 days | 09/07/2021 | 09/08/2021 | Project Manager |
| 2 | Design and Requirement Analysis | 4 days | 09/09/2021 | 09/12/2021 | Project Manager |
| 3 | Resource Planning and Hiring | 4 days | 09/12/2021 | 09/16/2021 | Project Manager |
| 4 | Development and Unit Testing | 55 days | 09/17/2021 | 11/10/2021 | Developers |
| 5 | QA Testing | 10 days | 11/11/2021 | 11/21/2021 | QA Testers |
| 6 | Bug Fixes | 7 days | 11/22/2021 | 11/29/2021 | Developers |
| 7 | User Acceptance Testing | 7 days | 11/30/2021 | 12/07/2021 | Developers, Project Manager |
| 8 | Product Delivery | 1 day | 12/08/2021 | 12/09/2021 | Project Manager |

The below table shows the estimated overall cost of the project:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Resources | Quantity | Cost | Time Utilized | Total Cost |
| Project Manager | 1 | $100/hour | 520 hours | $52,000 |
| Developers | 6 | $50/hour | 520 hours | $156,000 |
| QA Testers | 2 | $50/hour | 520 hours | $52,000 |
| Health Insurance of Team | 9 | $2000/resource |  | $18,000 |
| Hardware | 7 Laptops, 3 Desktop, 7 Mobile Devices, Storage Devices | $18000 |  | $18,000 |
| Software Cost | All around automation PL/SQL Developer, Jenkins, Jira, GitHub | $2000 |  | $2,000 |
| Building, Utility | 1 | $15000 |  | $15,000 |
| Payments Gateway | 1 | $30/Month | 4 months | $120 |
| Database and Cloud Servers | Oracle DB with backup servers, AWS | $7000/Month | 4 months | $28000 |
| Total |  |  |  | **$341,120** |

**Cost Estimation**

All the resources are utilized for 8 hours a day, working for 5 days a week. They are estimated to work for 520 hours in the entire duration of the product development.

The total estimated cost for the project is **$341,120.00**. Adding up a profit margin of 60% we will be quoting **$540,000.00** to the university.

# 7. Conclusion and Recommendations

This feasibility study gives the scope and resources needed to design and develop a web application and an android application. The estimated delivery time is 3 months, that covers all the required functionalities. In future, additional features can also be added depending upon the number of students using the system. The UFS once implemented and comes into usage, students would be hugely benefited. Students can order and have the meal at the comfort of their residence and reduce the waiting time at the restaurants. This website and the mobile app also increase the revenues of the restaurants.

The product has certain risks:

* Additional two resources required to test the functionalities in each sprint during the development phase. So, four developers and two QA testers required.
* Considering that payment gateway providers integrate the functionalities in the system on time. Else there will be less time span to test the orders functionalities and fix any bugs.
* The downtime or crashing of Database, Servers could further delay the planned progress and that would reflect either in increased cost by hiring more resources to deliver on time or deliver only primary functionalities.

Some benefits of the application are:

* The revenue of the restaurants will increase as more students can order food easily during peak dining hours.
* Students who are not able to visit restaurants can have it in the comfort of their home.
* The application is free to use, and other benefits like referral bonus, credits if ordered repeatedly will give students a discounted price when compared to offline orders.

The recommendation we provide here is that there is more scope of improvement with the customer experience with new features. Once after analysis of student satisfaction after the application is live, it could be planned, and new functionalities could be added in future.

# Appendices

AWS Elemental Servers can be used to host the web application. The recommended hardware requirement if Virtual Machines are used is:

For AWS Elemental Conductor Live 3, AWS Elemental Conductor File, AWS Elemental Statmux

RAM: 16 GB

Disk space: 500GB

CPU cores: 24

Processor speed: 2.3 GHz or more.

# References

1. <https://nakedsecurity.sophos.com/2013/11/20/serious-security-how-to-store-your-users-passwords-safely/>
2. <https://shiyamtj.wordpress.com/2018/09/26/requirement-stability-index/>
3. <https://www.simplilearn.com/project-and-process-metrics-article>
4. <https://online.visual-paradigm.com/drive/#diagramlist:proj=0&new=SystemContextDiagram>