THUTO KE THEBE

UNIVERSITY of BOTSWANA

DEPARTMENT OF SCIENCE
CSI 341: SOFTWARE ENGINEERING

PROJECT PLAN FOR UB PIZZERIA SYSTEM
BY: GROUP 9

APRIL 07 2021
DELIVERABLE 1

Authors:
Botlhe Tseme : 201802398
Lesego Kebokilwe :201801524
Lesego Nkomazana :201800476
Bhekisisa M. Mkhonta :201700590
Kealeboga A. Morwaagole : 201700246

Contents

Table of Figures

INTRODUCTION

UB Pizzeria is a business solely based on producing and distributing Pizza in and about Gaborone .The current system of ordering pizza used by UB Pizzeria University of Botswana is a manual system in which the services are in person however this way of providing services has been outpaced as the city and the university population grows and not forgetting the Covid-19 virus .This manual system tends to be tedious seeing as customers have to make their way to the UB Pizzeria in order to buy their pizza of which they place spend a lot of time queuing and even going to an extent of missing lessons and also at a risk of contracting the virus and some salesperson are not patient enough to listen to people when making orders . At times, the queue is not processed quickly enough for example, in the event of few salesperson the queue will move slowly wasting people's precious time. This older system is often found to be ineffective seeing as it is time-consuming, and poses health risks in the era we are living in. With the above situation software that customers can use to make orders, receive deliveries and many more services is urgent and can help Pizzeria management going forward. To bring this solution to life the waterfall methodology will be implemented thus aiding Pizzeria to move from a traditional to a contemporary way of business.

ROLES

To create this software, we will have:

- Front-End Web developers – This lot is responsible for the interfaces that will be used by the customers. HTML and CSS will be the most prominent languages used, of course not disregarding other languages that can and may be used.
- Back-End Web Developers – This lot will be responsible for "behind the scenes" part of the project. SQL & PHP will be used.
- Tec Leader- Merges the whole project together and oversees the whole project.

Software Development Methodology

Waterfall Development

We have chosen to go with the Waterfall methodology which will mean creating separate focus teams. These teams will work on different sequential stages (requirements, design, implementation, verification, deployment, and maintenance).

Following a traditional development method that relies on a rigid linear model. Each of the stages must be 100% complete before the next stage can start. Just like a waterfall fills lower-level pools, the stages of the Waterfall model flow from one to another. Meaning the method does not allow for going back to modify the project or direction.

Advantages

- The model is easy to understand and manage.
- The method is suitable for projects that are smaller in size and whose requirements can be determined upfront.
- The Waterfall development is recommended for less experienced project managers and project teams whose composition changes frequently.

Disadvantages

- The Waterfall method finishes one stage before the next one can begin. So, it is impossible to go back and edit something.
- It is not flexible and does not handle project risks well.
- It is not a good model for complex and long-term projects.

Figure below shows the flow of a waterfall model:

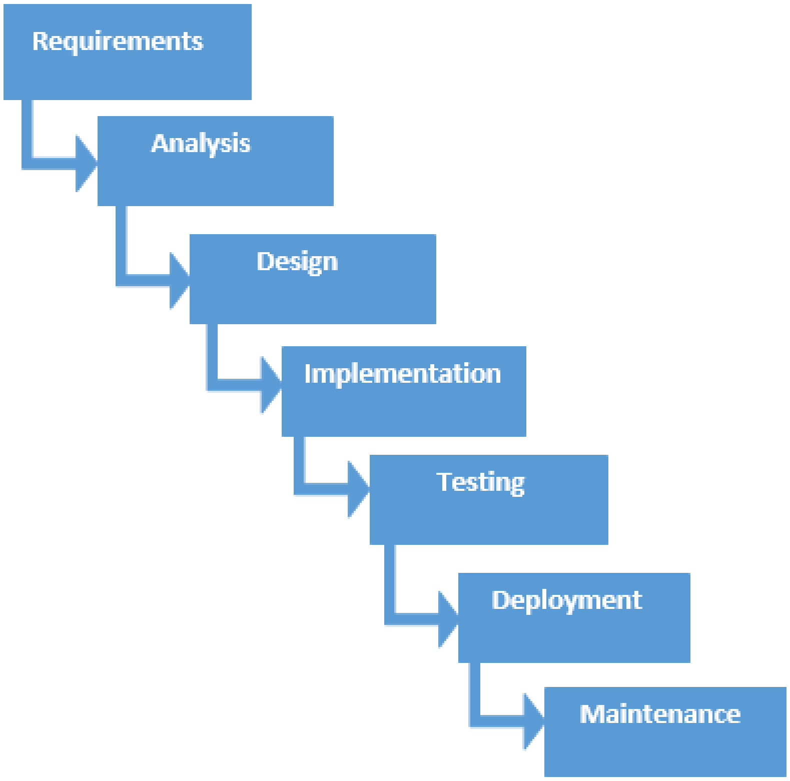


Figure 1: Waterfall Model

Data Gathering Methodology

The Questionnaire methodology will be used to further collect data requirements from future customers of UB Pizzeria for a more accurate gauge at the specifications required for our website application interface:

SYSTEM OUTLINE GOAL, OBJECTIVES, SUBSYSTEMS

SOFTWARE SYSTEM PURPOSE (GOAL)

Our goal is to provide a system that will be of ease of use and available most of the time, and much quicker, providing customers and salesperson with the opportunity to manage orders and place orders report by enabling users of the system to access an online UB Pizzeria web-portal from anywhere in the university premises for a better and conducive buying environment without queuing for a long time and risking their lives to contracting covid 19.

SOFTWARE SYSTEM OBJECTIVES

- To assist customers when ordering their pizza from the UB Pizzeria without going directly there
- To allow customers to take their time and customize their orders and fulfill their desires and not buying whilst they are hurried
- To help customers keep track of their orders after purchasing within the 30 minutes time given.
- To help UB pizzeria management to process orders in time and deliver goods to customers swiftly and easily.
- To help UB Pizzeria management to view orders and keep check of how many sales they are doing per day and ways of improving them

SUBSYSTEMS

Database subsystem: this subsystem will store the details of the customers / admin / salesperson /orders, such as order no and payments details .

Authentication subsystem: this subsystem will check to see if the user who is logging into the system is allowed to access the system and will check whether that user is an administrator or customer

Ordering subsystem: this subsystem will allow the user of the system to place an order and customize their orders.

Tracking subsystem: this subsystem will allow the user of the system to check if their order has been processed yet or not.

Cart Subsystem: this subsystem allows the user to view all the items in case he or she would want to buy many items at a go.

Administrator subsystem: this subsystem allows for the admin managing the accounts of the regular users of the system that is, adding a user, remover a user, editing a user's account and changing a user's password and updating order status adding products to the system and the stock.

Google Maps subsystem: this subsystem allows the user to view all he or she is to the UB Pizzeria and get his location with relation to UB Pizzeria.

Payment Subsystem: this subsystem allows the user to pay online if he or she does not want cash on delivery (COD).

Chabot subsystem: this subsystem allows the user to ask questions and get feedback instantly if all the salesperson are occupied.

Registration subsystem: this subsystem allows the users to create their accounts and register unto the system to use it.

PROJECT MANAGEMENT
WORK BREAKDOWN STRUCTURE (WBS)

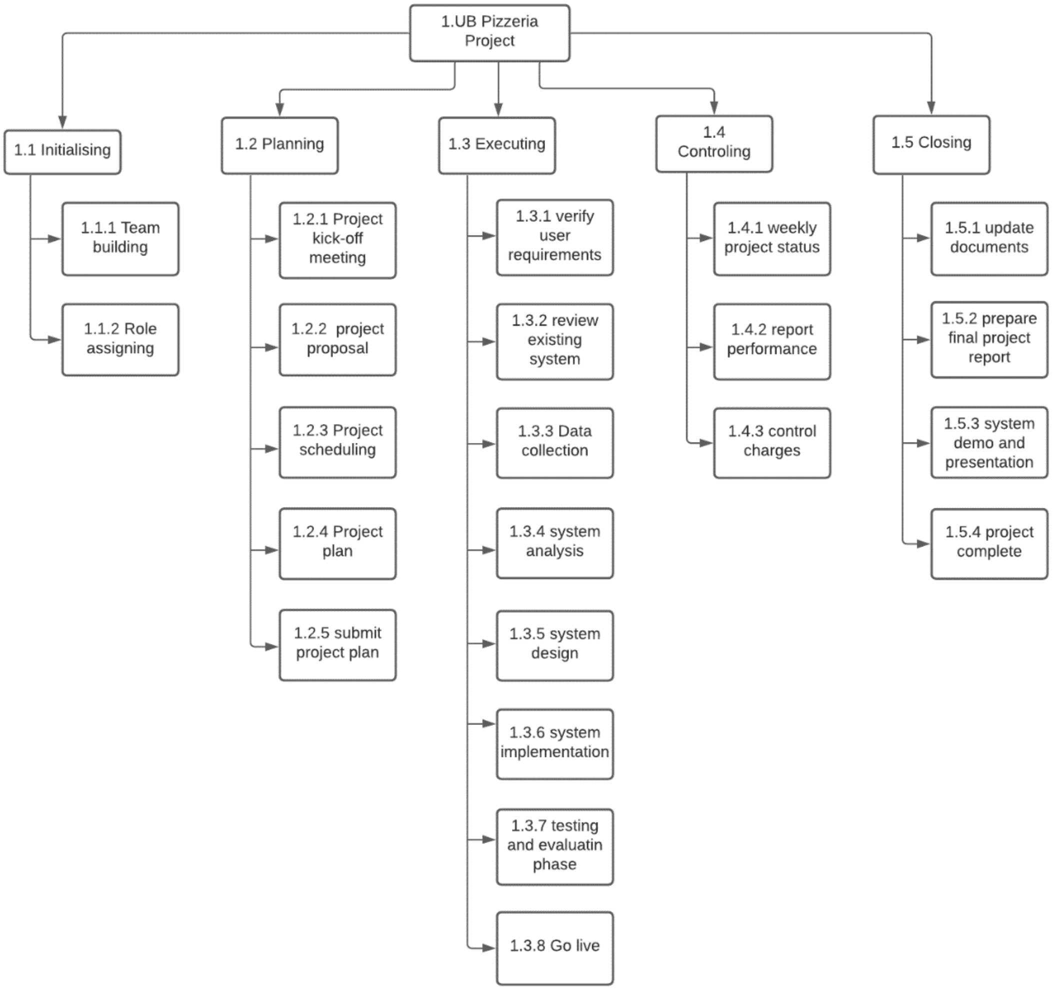


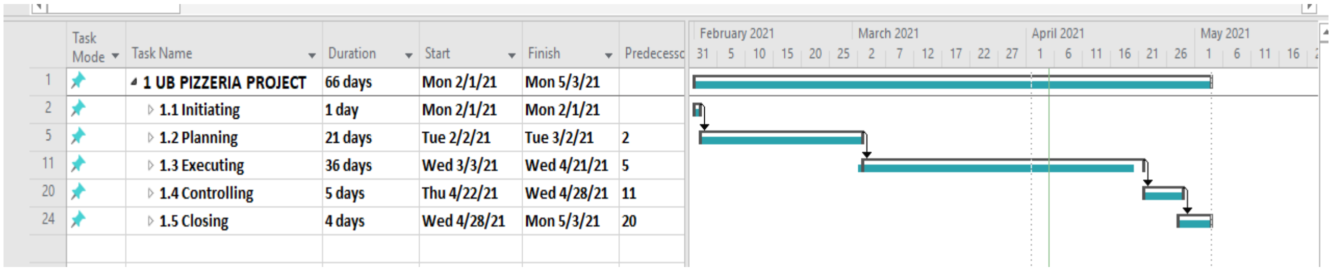
Figure2: Work Breakdown Structure

HARDWARE AND SOFTWARE RESOURCES

- To carry out the proposed project, the resources intended for use minimally include, but are not limited to:
- Time (of the 5 group members involved in the group project)
- Computer Devices (provided by each of the members, the project shall be conducted in a bring your device environment)
- Programming Language IDE (required developing the system in the soon-to-be-decided programming languages).
- Funds BWP 40.00 to buy a domain and host the system.

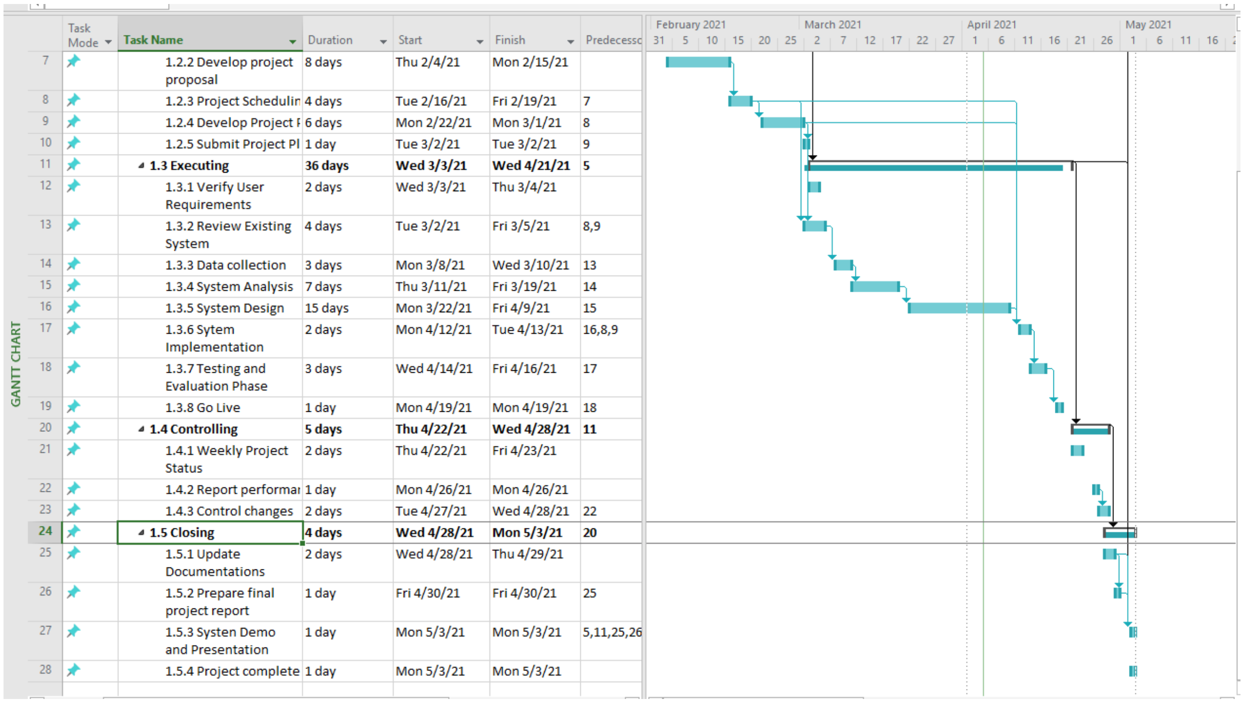
PROJECT TIME PLANNING

Below are figures of the project time plan as to be carried out:



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1	📌	◢ 1 UB PIZZERIA PROJECT	66 days	Mon 2/1/21	Mon 5/3/21	
2	📌	▷ 1.1 Initiating	1 day	Mon 2/1/21	Mon 2/1/21	
5	📌	▷ 1.2 Planning	21 days	Tue 2/2/21	Tue 3/2/21	2
11	📌	▷ 1.3 Executing	36 days	Wed 3/3/21	Wed 4/21/21	5
20	📌	▷ 1.4 Controlling	5 days	Thu 4/22/21	Wed 4/28/21	11
24	📌	▷ 1.5 Closing	4 days	Wed 4/28/21	Mon 5/3/21	20

Figure 3: Gantt Chart of Major Activities/Milestones



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
7	📌	1.2.2 Develop project proposal	8 days	Thu 2/4/21	Mon 2/15/21	
8	📌	1.2.3 Project Schedulin	4 days	Tue 2/16/21	Fri 2/19/21	7
9	📌	1.2.4 Develop Project F	6 days	Mon 2/22/21	Mon 3/1/21	8
10	📌	1.2.5 Submit Project Pl	1 day	Tue 3/2/21	Tue 3/2/21	9
11	📌	**◢ 1.3 Executing**	**36 days**	**Wed 3/3/21**	**Wed 4/21/21**	**5**
12	📌	1.3.1 Verify User Requirements	2 days	Wed 3/3/21	Thu 3/4/21	
13	📌	1.3.2 Review Existing System	4 days	Tue 3/2/21	Fri 3/5/21	8,9
14	📌	1.3.3 Data collection	3 days	Mon 3/8/21	Wed 3/10/21	13
15	📌	1.3.4 System Analysis	7 days	Thu 3/11/21	Fri 3/19/21	14
16	📌	1.3.5 System Design	15 days	Mon 3/22/21	Fri 4/9/21	15
17	📌	1.3.6 Sytem Implementation	2 days	Mon 4/12/21	Tue 4/13/21	16,8,9
18	📌	1.3.7 Testing and Evaluation Phase	3 days	Wed 4/14/21	Fri 4/16/21	17
19	📌	1.3.8 Go Live	1 day	Mon 4/19/21	Mon 4/19/21	18
20	📌	**◢ 1.4 Controlling**	**5 days**	**Thu 4/22/21**	**Wed 4/28/21**	**11**
21	📌	1.4.1 Weekly Project Status	2 days	Thu 4/22/21	Fri 4/23/21	
22	📌	1.4.2 Report performar	1 day	Mon 4/26/21	Mon 4/26/21	
23	📌	1.4.3 Control changes	2 days	Tue 4/27/21	Wed 4/28/21	22
24	📌	**◢ 1.5 Closing**	**4 days**	**Wed 4/28/21**	**Mon 5/3/21**	**20**
25	📌	1.5.1 Update Documentations	2 days	Wed 4/28/21	Thu 4/29/21	
26	📌	1.5.2 Prepare final project report	1 day	Fri 4/30/21	Fri 4/30/21	25
27	📌	1.5.3 Systen Demo and Presentation	1 day	Mon 5/3/21	Mon 5/3/21	5,11,25,26
28	📌	1.5.4 Project complete	1 day	Mon 5/3/21	Mon 5/3/21	

Figure 4: Gantt Chart of Major and Minor Activities

CASE TOOLS

Upper Case Tools

To be used for the planning, analysis, and design of the proposed software system.

- Documentation tools: Microsoft Word
- Planning tools: Microsoft Project
- Modelling tools: Microsoft Visio
- Lucid Chart
- Google Maps api
- Google Online payment api
- Design – Hypertext Preprocessor (PHP), HTML, CSS, Javscript, boostrap, Ajax, Jquery, Javascript

Lower Case Tools

To be used in the implementation, testing and maintenance of the proposed software system
-Programming tools such as an Integrated Development Environment: Visual Studio Code
-Data repository: MySQL database

RISK MANAGEMENT

Most program building ventures are hazardous since of the rum of the genuine potential issues that can emerge. The essential good thing about hazard administration is to contain and mitigate dangers to extend victory. You have got to distinguish and arrange, and after that be ready to act when a chance arises- drawing upon the encounter and information of the whole group to play down the effect to the extend.

Program hazard administration incorporates the distinguishing proof and classification of specialised, automatic and handle dangers, which gotten to be portion of a arrange that joins each to a relief technique. The extend supervisor (Project Manager) screens chance amid the extend. If any materialise, a particular proprietor actualises a moderating activity. Hence, we come up with a viable computer program hazard administration being our risk management plan.

Risk	Risk Description	Probability	Risk Impact	Contingency Plans	Assignee
Technical	This includes risks based on requirements, the technology being used, interfaces, performance, and quality. Arise when only a part of the developer's team is familiar with the software: · Endless changes of requirements for the software · Existing technology required for the development is only in its initial stage with no advanced technology being available. · The project is too complex to	Medium	Tasks not being completed on time.	· There is no need to focus too much on popular technology. Instead, you need to specify the problem you're trying to solve & choose a technology stack that deals better with that problem. · make sure all the tools have proper and updated specifications. Add a decent buffer to the estimate in case there are updates developers haven't worked with yet. ·Use at least a	Analyst

	implement. • Integration of modules turns out to be too much of a challenge within the project performance.			basic specification of new technology. •Request all the existing documentation, try to investigate the issues that complicated the work earlier. Take every chance to speak with the previous team and discuss the details. It may make sense to perform source code analysis before you deliver an estimate, as you'll never know what challenges you'll meet afterward.	
Management	The risks that come up from planning, scheduling, estimating, or communication. This concept includes: • poor communication and interaction within the team; • insufficient qualification level of PM; • lack of leadership and analytical skills; • poor risk management in software	Medium	Difficulty in getting started	• When estimating the project, add enough design hours to conduct product workshops, UX strategy, UI design, and usability testing. •If both part-time projects are long-term, it makes sense to allocate a back-up specialist. •additional coordination. It makes sense to put a project on	Tec leader

	development.			hold and accumulate a reasonable number of tasks if the workload is less.	
Organizational	Risks that any project dependencies, logistics, resources, budget and productivity. This includes: • poor project management; • incorrectly chosen methodology, Very generic specification. · incorrectly matched team members. • the budget was initially calculated incorrectly; • no funds have been reserved; • unplanned project expansion. • Incorrect deadlines set by a client. · The client requires too much communication.	Low	The team being overwhelmed by the workload hence inability to focus.	· Before starting a project, the development team should inform the client about the high probability of a risk event. If the deadlines cannot be moved forward, it makes sense to focus on the most important features instead of spreading efforts to each task. ·It’s better to start with business analysis and convert the requirements into the functional specification where all the features are described in detail and prioritized by their importance.	Tec leader and analyst
External	These are risks that come from your customers, users, contractors, or even the market itself.	Low	Difficulty integrating work.	·Research the market, country of operation, and worldwide tendencies to analyze the	Marketing manager

	They are dangerous due to their unpredictability. They include: ·Limited funds for continuous development of the project ·Market's rapid development ·Inevitable changes in customer product strategy and priority ·Government rule changes			current situation. ·discern the patterns on the market and make sure that they are favourable for the product and business model.	

PARTICIPATION AND CONTRIBUTION OF EACH GROUP MEMBER

System Outline Goal & Objectives:

Botlhe Tseme
Lesego Nkomazana
Bhekisisa Mkhonta
Lesego Kebokilwe
Kealeboga Morwaagole

Case Tools:

Botlhe Tseme

Subsystems:

Lesego Nkomazana
Bhekisisa Mkhonta
Lesego Kebokilwe
Kealeboga Morwaagole

Project Management:

Kealeboga Morwaagole

Risk Management:

Lesego Kebokilwe

Methodology:

Bhekisisa Mkhonta

Introduction and roles

Lesego Nkomazana

The team members contributed successfully.

REFERENCES

- Benington, Herbert D. (1983). "Production of Large Computer Programs". IEEE Annals of the History of Computing.