FAST FOOD APPLICATION

CST-17 SOFTWARE ENGINEERING COURSE PROJECT: AN ONLINE CANTEEN FOOD ORDERING APLICATION

Group Leader: Margaret Mwewa F17040112

Group Members: Raymond Kuree F17040120

Kelvin Asare F17040111

Lukundo Kampeshi F17040109

Group Name: LKMR

Instructor: Liu Zheng

Due Date: 2019-12-17

TABLE OF CONTENTS

DOCUMENTATION FOR USERS

As a group we decided to make a web application designed to order food in our school canteen. Web applications are applications that are deployed via the internet, and run within the browser itself. The application is hosted on the website's servers, rather than stored locally on your computer. This application is going to be built targeting smart phones and their users. We intend on making it desirable for both students and teachers to use seeing that they are the ones that access the canteen. Our goal is to have an application that can be used to browse through the different menu options, and allow you to order your food from outside the canteen. It will also have a provision that lets you pay from your phone and show you your order number and collection window.

When you go to the canteen you'll have to first decide on what you'd want to eat; therefore, you'll have to go from one desirable window to the next looking for what you might like. Doing this takes a bit of time and not only that, you will then have to queue up to place an order and then wait for the order to be prepared. The whole process up until you get your food will take time, and this further causes another problem, a crowd.

Having a large group of people in the canteen queuing up and standing around can lead to an overpopulated canteen. Once this happens, there will be fewer sitting spots, many people standing around, noise pollution, air pollution, to mention a few etc. etc. Paying for your food in the canteen is done via student card. In the attempt to eat fast and leave room for other students and teachers to sit down teachers and students are at a risk of leaving their student cards behind as we see this happening on a daily basis.

DESCRIPTION

The application is centred on targeting smart phone users' students and teachers alike.

The basis for the application is to access the canteen menu from the comfort of their dormitory without glitches. Basically this application is going to allow the easy

Patronization of food, it is going to save time and energy of the teachers and students.

Furthermore, it will reduce the congestion situation in the canteen.

HOW TO USE THE SOFTWARE

This user interface of the web application is relatively easy to manoeuvre.

Here is an outline of the basic steps:

1. Visit the home page "Welcome to FAST FOOD"

2. Click Login/Register located at the top right of the page then fill in and submit the form to sign up for a new account.

3. After successful registration, the web application will be redirected to the login page. The user must fill in the login form with their username and password.

4. The user is now logged into the system, and is redirected to the “Place your order” page. The user can now select the desired food and quantity and click confirm order.

5. The order has now been placed successfully and the user must pay the stated amount upon collection of their food.

Currently there is no way for a user to report a bug.

DOCUMENTATION FOR DEVELOPERS

HOW TO OBTAIN THE SOURCE CODE?

Firstly, let us talk about how and where to find our repository on GitHub. With the assumption one is really not familiar with GitHub, they can go to GitHub.com and then sign up for a new account, however this step is not necessary. Once they have gained access to GitHub they can search for whatever software information they are looking for. To find the repository in which FAST FOOD project is located, they would need to search for the user njuptcourses. Within njuptcourses there are presently 16 repositories. FAST FOOD is located in se-2019-autumn-lkmr. In se-2019-autumn-lkmr there is a zip folder "FAST FOOD", this folder contains all the necessary php files, background images, menu images, css and js files needed to execute the WebApp. In addition to the zip folder the previously mentioned files are also uploaded in the repository of se-2019-autum-lkmr.

LAYOUT OF THE DIRECTORY

Within se-2019-autumn-lkmr there are **five documents in PDF format** namely LKMR_SDS_slides, LKMR_SDS_doc, LKMR_SRS, fastfood_doc and fastfood_slides. These five documents are the software design specifications presentation and document, software requirements specification document, project proposal document and presentation, respectively.

There is also **one 'css' folder** containing the main.css file**, one 'js' folder** containing the jquery.validate.min.js, jquery-1.82.min.js and main.js files, **one 'menu' folder** containing thirteen images in jpg format, **one fastfood.sql file**, **three background images** namely bg1.jpg, bg2.jpg, background1.jpg, **one style.css file** and **seven php files** namely config.php, header.php, index.php, help.php, order.php, login.php, register.php. There are **three more contents** in the directory, namely, the LKMR_FINAL_RELEASE_doc, LKMR_FINAL_RELEASE_demo and the FASTFOOD zip folder.

HOW TO BUILD THE SOFTWARE?

Prerequisites:

1. Download and configure Wampserver for Apache, Php or MySQL.

http://www.wampserver.com/en/

2. In the Wampserver folder, there is a subfolder 'www', unzip the FAST FOOD folder into the 'www' subfolder.

3. Once Wampserver is running on your computer, and the icon on the taskbar is green. Select phpMyAdmin and login into the system.

4. Select ‘Databases’ and then ‘import’. Import the foodies.sql file contained in the FAST FOOD folder in the ‘www’ folder of Wampserver.

Once the configuration is done, you can access the web application by visiting this url: http://localhost/FAST%20FOOD/index.php to access the home page of fast food.

To develop the application further or make changes, the developer can use a text editor such as notepad to modify or create php, html and css files. However, to update the database or create new tables and queries in the database, the developer would find tools such as MySQL Workbench helpful and easy to use. It is also possible to manipulate the database directly from phpMyAdmin provided by the Wampserver. To implement the changes and update the website, the developer just has to save all the changes made.

HOW TO DEBUG SOFTWARE

The first step in debugging our web application is to test it thoroughly. Using every feature to make sure that everything works as expected.

One of the key steps to debugging web applications actually begins before deployment. By thoroughly documenting the code, it is easier to find offending sections that caused issues during your features testing.

Web browsers such as Google Chrome provide additional features that aid in debugging through the use of browser extensions. The Web Developer Toolbar is one of several such extensions. However, browsers like internet explorer can only run the application and look for crashes.

It is also possible to make use of most Integrated Development Environments (IDEs). IDEs will have a debug feature, which will look for syntax and grammatical errors that can cause code to function erratically.

Here are some steps that can be followed to aid the debugging process:

1. Always reproduce the bug before changing code.
2. Write a test case that reproduces the bug.
3. Know the error codes.
4. Utilising two or more developers to separately debug and run the code.
5. Defect Analysis using modelling, documentations, finding and testing candidate flaws, etc.
6. Defect Resolution by making required changes to the system and validation of corrections.

As LKMR we submitted an SRS schedule with an approximate time frame for how we thought our team would spend its time. The SRS document has been updated to a new schedule showing how each member actually spent their work time. Below is a list of the tasks we accomplished and the amount of time we thought we'd spend versus the actual time we spent.

1. Project Proposal
   Delivering our project proposal was the first thing we did.
   - Brainstorming of project ideas: Carried out by the four members of the group, we gave ourselves 7 days and it actually us approximately 5 days.
   - Project proposal (vision, software architecture): Estimated to be carried out by Margaret and Lukundo for 7-14 weeks. Lukundo came up with and wrote the vision, software architecture and challenges and risks. Margaret provided the pictures for the PowerPoint presentation and designed the user interface. All this was actually done within 12 days.
   - Finally, Kelvin presented the PowerPoint presentation as a completion of assignment one.

2. Software Requirement Specifications
   We aimed at completing the software requirement specification document deliverable within o7-14 days.
   - UI diagrams: The user interface diagrams were provided by Margaret in a period of 6 days.
   - Product Description: The product description was provided by Lukundo within 7 days.
   - Use cases: Margaret and Lukundo came up and wrote the use cases within 7 days.
   - Process: Margaret and Lukundo came up with and wrote the process aspect within 4 days. This concludes assignment 2 and there was no presentation for it; only uploading of the SRS document.

3. Software Design Specification deliverables
   A time frame of 14 days was set aside to complete and provide the software design specification document and PowerPoint presentation.
   SDS PowerPoint presentation:
   - Design: Raymond designed and put together the presentation within 7 days of the estimated 14 days.
   - Presentation: Margaret and Lukundo conducted the presentation.

   UML Diagrams:
   - System diagram: This was done by Lukundo within 4 days of the estimated 7 days.
   - Class diagram: Margaret and Raymond, within 7 days of the estimated 14 days.
   - Sequence diagrams: Margaret and Kelvin within 6 days.

   SDS Document:

- Architecture section: The system architecture was provided by Lukundo, within 6 days.
- Process section: The process section, team structure and coding style guidelines were provided by Margaret in 5 days.

FEATURES AND CUTS

The major functionalities that we aimed at for our web application as listed in our SDS document were to register to use the application; choose a canteen; browse through the list of food; pick what you want; select whether it is takeaway or eat from the canteen and pay online. After which you will wait to receive an order window to pick your food from. We managed to complete the major functions of the app though we couldn't manage to complete a few of them. In our SRS document we stated that the admin can be able to check the database and see everything that happens in the canteen, delete a registered user and change their information. We did not manage to do the part that lets the admin to delete a user from the database. In our document we didn't state that you would be able to log out of the account because once a student has logged in there won't be any need for them to log out. If there's need for someone else to log in on the same phone a user can click on the button that says 'Register/Login', this will enable another user to register for the app. We couldn't do the part of the app that allows you to pay for the ordered food online. Trying to fix this was going to cost so much time so instead the system was changed to a system that lets you pay by cash. We also couldn't manage to get to the page that shows which order window you are to collect your food from after you order. So we decided to display the order window number after the application tells you how much you are to pay. We chose to cut this feature because we'd have to have a whole online banking system functioning and this would've taken us as long as a month to get this done. So by changing it to a cash payment system we cut out almost 14 days of work.

TASKS ASSIGNMENTS AND COMPLETION

NAME	CURRENT ROLES	MOST TIME SPENT	LEAST TIME SPENT
Margaret (Group Leader)	-Provided the pictures for the PowerPoint presentation -Designed the user interface -Came up and wrote the use cases; wrote the process aspect - Conducted the second presentation - Class diagram and Sequence diagrams -Creating the web app	The most time spent was when making the application. It took about 7 days to work on. It took the longest to complete because she had to get familiar with wampserver and SQL and debug and fix any problems with the program	The least time on taking pictures of the canteen for the first PowerPoint. She spent the least time on this task because she only needed to go the canteen with her camera during peak hours to take pictures.

Lukundo (Group Member)	-Came up with and wrote the vision, software architecture and challenges and risks -Came up and wrote the product description; use cases; wrote the process aspect -Conducted the second presentation -System diagram and system architecture -Requirements and schedule postmortem	The task which took the longest was writing the requirements and Schedule Postmortem. It took the longest because information was needed from the group members when everything was done. This information about the members individual tasks and how long they took once everything was completed needed to be put in the schedule postmortem	The task which took the least time was conducting the presentation. This was done within 10-15 minutes. It took the least time in her opinion because she only presented two slides in the ppt. The rest were explained by another student member.
Kelvin (Group Member)	-Presented the PowerPoint presentation -Sequence diagrams -Documentation for users and developers	Documentation for users and the documentation developers took the longest time. It took long because it was a 5 paged document that describes the users and the developers' documentation, i.e. how the application works from their view and what they are to do. He worked hand in hand with another group member.	The task which took the least amount of time was presenting of the first ppt which was completed within a period during class.
Raymond (Group Member)	-Designed and put together the presentation -Class diagram -Documentation for users and developers	Documentation for users and the documentation developers took the longest time. It took long because it was a 5 paged document that describes the users and the developer's documentation, i.e. how the application works from their view and what they are to do. He worked hand in hand with another group member.	The task which took the least amount of time was drawing the class diagram. It took the least amount of time because he did it together with another group member.

GROUP MEMBERS TASKS ASSIGNMENT AND COMPLETION REPORT

1. F17040111 KELVIN ASARE

First task was the brainstorming of project ideas. It was a group idea to develop this web application. It took us some time to come up with what we really wanted to develop because it was a risk in the first place, we had to be sure the task we are giving ourselves will be able to be achieved by the end of the day. So finally we came up with the idea of creating a food ordering website to help both students and teachers to order food without having to be in a long queue and being able to pay online to avoid using student cards. Also my second task was to draw a sequence diagram for use case for food order. I did that with an online website called “draw.io”. It wasn't equally an easy task but I managed to get it done. My next task was to prepare a design presentation ppt. I did that with the help of my other group member, F17040120- RAYMOND KUREE. Now it was time for the final project, we had to divide the task because there was just a limited time to deliver the software/application. My task was to deliver a 4-page group document about documentation for users and documentation for developers. I had my information from the help of internet. My other group member (F17040120-RAYMOND KUREE) helped me with the documentation for users.

2. F17040120 RAYMOND KUREE

The first task was joining ideas on what to do. We all came up with the idea of building this application. We really took time on what to develop but at the end, we took the risk. we had to be sure we were going to achieve our goal and be able to build it up. The whole application is about creating a food ordering website to help both students and teachers to order food without having to be in long queues and be able to pay online and avoid using student cards. My first task as an individual in the group was to handle the class diagram. I was a little bit confused with it but I got help from a team member (F17040112-MARGARET MWEWA). It wasn't an easy task but it got done at the end. My next task was to deliver a four-page group document about documentation for users and documentation for developers. I had to split it with a group member (F17040111-KELVIN ASARE), so I happened to take the documentation for users. The internet really helped me on my part. I managed to pull the weight and deliver.

3. F17040109 LUKUNDO KAMPESHI

One of the deliverables for this assignment is a one-page task assignment and completion for each member. Below is a brief description of what were my roles in creating this web application were, what I managed to complete and what I didn't manage to complete.

Assignment 1

My role for the first assignment was to brainstorm for ideas as well as the other 3 people in my group. I also made the presentation that introduced our idea, what we intended for it to do and also how we were planning on making the web application. The presentation comprised of a vision; software architecture; challenges and risks and problems that will be solved if our web app was to run smoothly.

Assignment 2

In the second assignment my role was to write the product description for the Software Requirement Specifications (SRS) document. This part of the document gives a detailed description of the 'FAST FOOD APPLICATION', as we called it, and its functions. What it should and shouldn't do and how best it will do it. It further explains in detail the current problems being faced in the canteens and it gives alternative solutions through this app. Another role I had for the second role was to come up with use cases. Use cases are a documentation of assumed scenarios written to have an idea of what the application does when the user is using it. They briefly describe situations that users find themselves in. I also came up with wrote the process aspect for the SRS. Coming up with and writing the use cases and the process aspect was done by me and another group member. Presenting of assignment 2 was also done by me and another group member.

Assignment 3

For assignment 3 I had to draw a system diagram. A system diagram shows the connection between these modules being used in this system, namely: admin, user interface, canteen, database, client and bursar (for the online payment that we intended to do). I also had to write the system architecture for the food application. It is the conceptual model that defines the structure, behaviour, and more views of a system. It is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.

Assignment 4

For assignment 4, my role was to do the database for the application in SQL or SDS, but we changed the application being used and my role changed to writing the application requirements and schedule post-mortem.

4. F17040112 MARGARET MWEWA

As group leader, I took responsibility for numerous tasks and aiding other group members. My main responsibilities were setting schedules, allocating tasks and monitoring completion of tasks. My most significant responsibility was uploading all deliverable on to GitHub before the given deadlines. Below is a brief description of all tasks I completed:

First Assignment: Project Proposal

I participated in the brainstorming of project ideas and project names, this was a group effort. I also carried out research on the feasibility of proposed projects, possible software, skillset and approximate time required to successfully complete the web application project. This process took approximately one week. The first assignment also required a project proposal document and presentation, my task was to provide pictures from the school canteen to give a clearer picture of our project's basis. I was also responsible for designing the mock user inter face diagrams. It was a challenging and fun task and I was able to complete it through the aid of an online software "Mockflow".

Second assignment: Software Requirement Specification

With the software Requirement Specifications (SRS) document, I was responsible for providing the user interface diagrams. I also provided the group dynamics and Schedule/Timeline under the process part of the document. With the help of my group member, Lukundo, I came up with six use cases even though we only shortlisted four. I was also responsible for gathering all the information necessary and putting the document together.

Third assignment: Software Design Specification

The third assignment consisted of two deliverables: a software design specification document and a PowerPoint presentation. I delivered the presentation with a fellow group member. I aided a fellow group member and re-designed the UML class diagram. I was also responsible for drawing one UML sequence diagram. I was able to draw the UML diagrams with the aid of an online software. I was also responsible for the contents under the process section namely the risk assessment, team structure, milestones, communications and coding style guidelines.

Fourth assignment: Final Release

With the final assignment my initial set responsibility was to provide the source code of the web application, excluding the sql and provide the content of the requirements and schedule post-mortem. However, I ended up providing the sql database and all the html, php, css and js source code. I was solely responsible for the programming and testing of the web application. It was a great challenge as I had no prior knowledge on building a web app I occasionally consulted a senior student for help as well as spending a lot of time on the internet learning how to build a web application. My next task was to provide this personal task assignment and completion summary, and my final task was to provide the final release demo and present the demo of the web application during the final class.