

FAST FOOD APPLICATION

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

Group Leader: Margaret Mwewa F17040112

Group Members: Raymond Kuree F17040120

Kelvin Asare F17040111

Lukundo Kampeshi F17040109

Group Name: LKMR

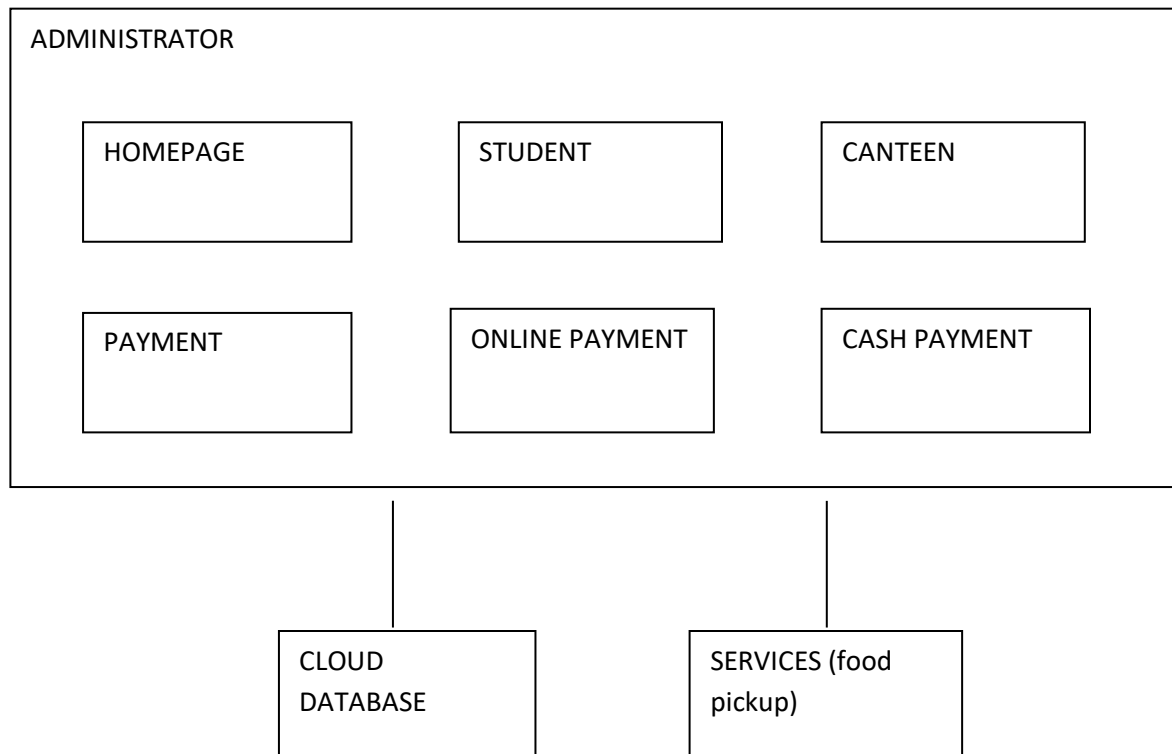
Instructor: Liu Zheng

Due Date: 2019-11-26

SYSTEM ARCHITECHTURE

At this point in the systems engineering life cycle, an operational need has been expressed and turned into a concept and set of operational requirements. A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. The application will have a graphical user interface and it will be user friendly. We will use HTML and SQL as our programming languages. One will open our application on their phone, there they can browse through the different menu options, select food, select a pick up time during the working hours, choose whether it's a sit down or take away. Below is a detailed definition of the systems software components:

Software components are designed to be reusable. They reduce extremely complex problems into small manageable problems. The difference between a costly, unstable, low performance system and a fast, cheap and reliable system often comes down to how well it has been architected into components.

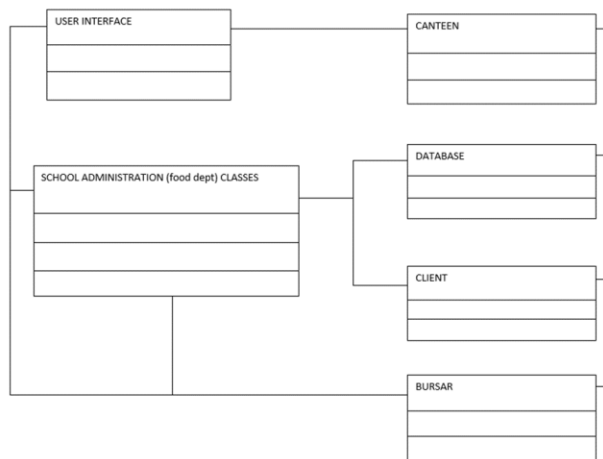


The connection between these modules is that when the homepage is opened, a login page appears for the student to log in. From a users point of view they will select what they want to eat; they'll pick what canteen they want to eat from, select whether they want to eat from the canteen or do a takeaway. They will choose whether to pay from their student card or use cash. If they use the student card, they will select whether they want to top up from their bank card if

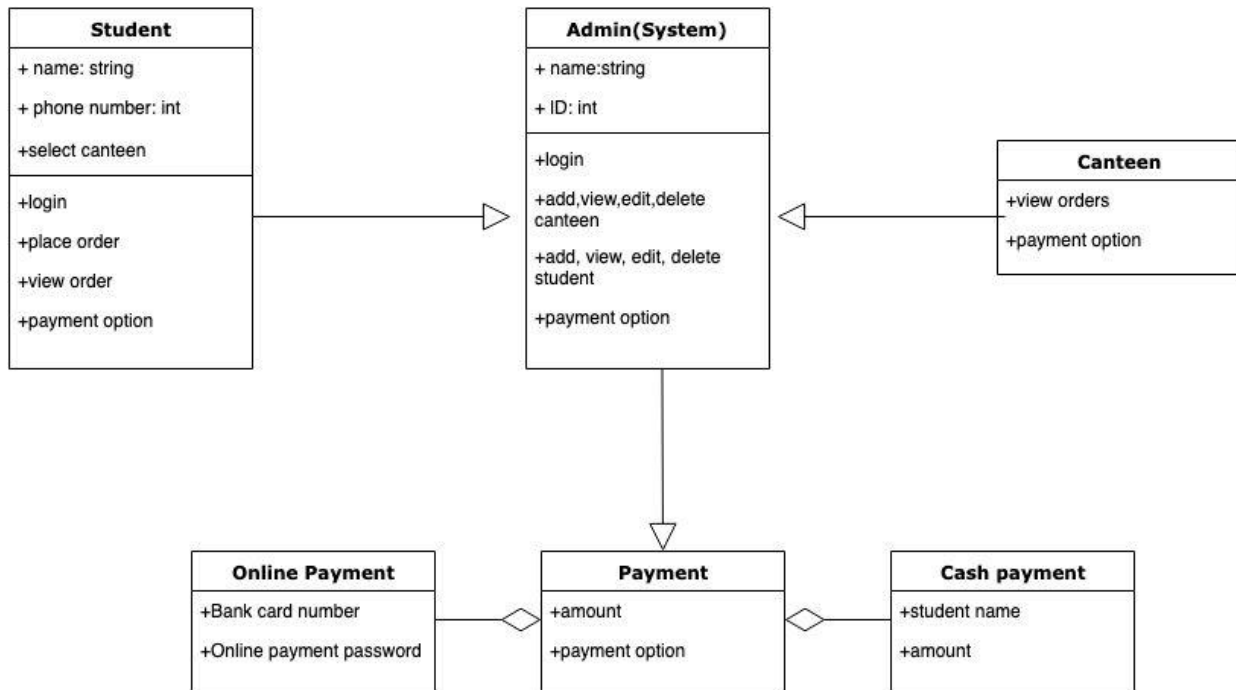
necessary. The people that serve in the canteen will also get the login page, when they log in, they will be able to see orders that they will have to prepare, they will also see whether the orders are to be packed as takeaway or put on a tray. From an administrator's point of view, when a student or a teacher orders food from the canteen application, the administrator can see what was ordered, who ordered what type of food and which window they ordered from. The administrator can also see the canteen's database: what they cooked, who ordered the food and whether they paid or not. The canteen's information and the student's information must match. From a developer's point of view, they must be able to see everything that goes on from the time the student, teacher or administrator logs in. They can see everything in the customer's database, canteen's database, and the administrator's database. The developer has access to everything and is able to change any part of the app at any time. They do not have a log in page but can check from the

Our data system first and foremost stores information of the people that will be able to access the canteen, both students and administrators. It will also have information on the available menus in the canteen, shows which menus have been registered as an order and which menus have been processed. Our database will have information on the student card and their corresponding bank card. We shall create a database using access (or something else) they will have a relation and will be linked to each other. This way it will be easier to retrieve information from each database and relate it to the next one.

SYSTEM DIAGRAM

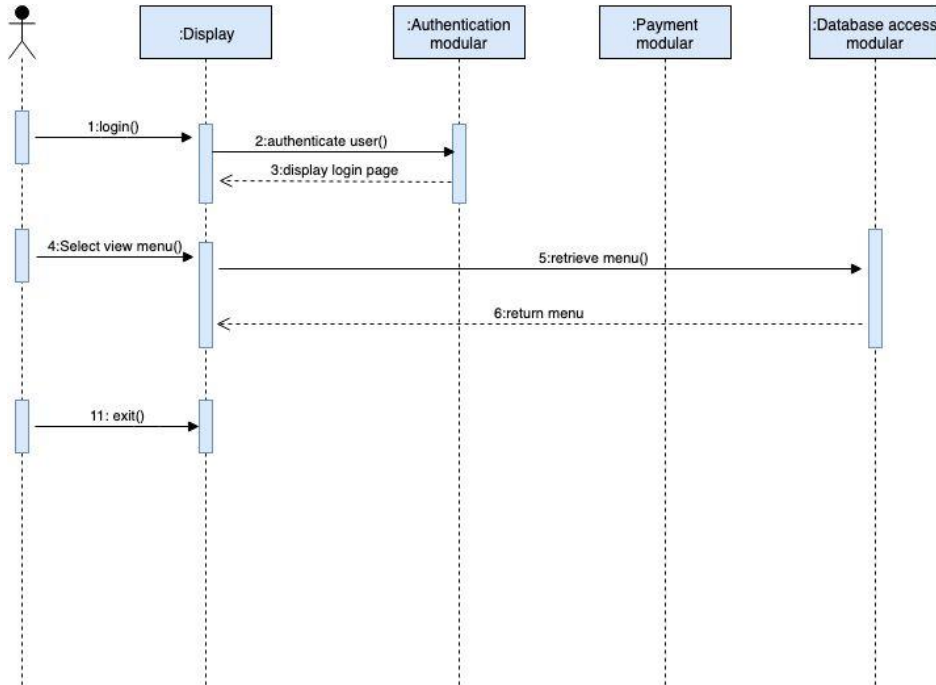


CLASS DIAGRAM

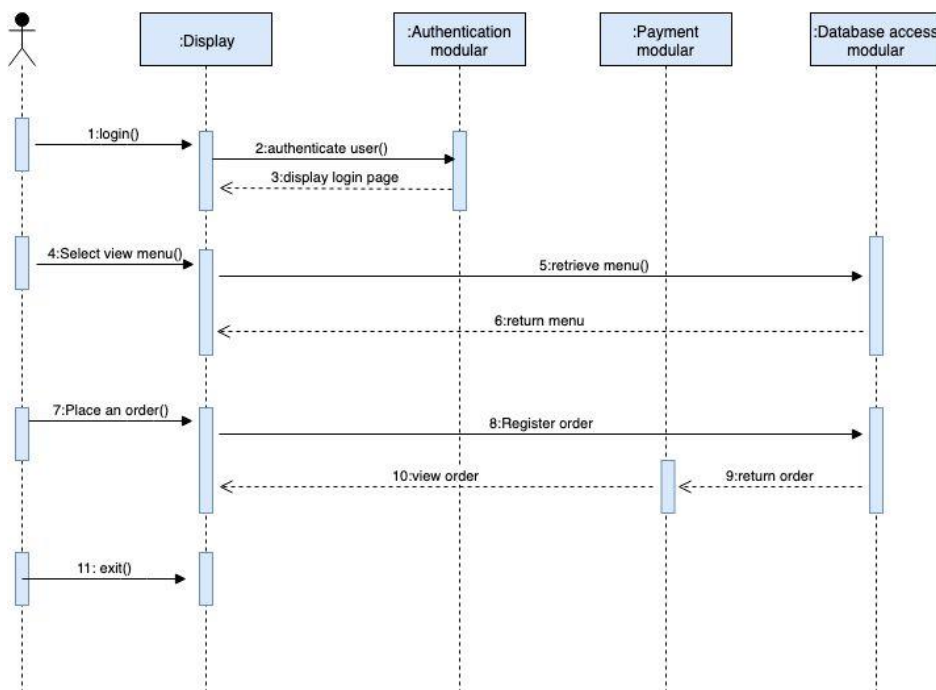


SEQUENCE DIAGRAMS

View menu sequence diagram:



Place order sequence diagram:



PROCESS

RISK ASSESSMENT

	RISK 1: Poor Leadership	RISK 2: Schedule Risk	RISK 3: Legal Risk	RISK 4: Operational Risk	Risk 5: Insufficient skill set
Likelihood of occurring	Medium	High	Medium	Medium	Medium
Impact (if occurred)	High	High	High	High	High
Evidence upon which estimates are based	The leader tends to be not vocal enough and ends up trying to complete tasks by herself.	Constantly falling behind schedule and trying to complete tasks at the last minute.	Collecting, storing and using personal information requires special permissions.	Schedule and leadership risks would affect the development and implementation of the project.	Basic knowledge of coding, but not in depth to a point where we can write an application without assistance.
Steps taken to reduce impact/likelihood	Alternating roles to establish who is a more suitable leader.	Establishing a set schedule and division of labor.	Researching on related laws and policies.	Reducing the likelihood and quickly mitigating other risks.	Identifying strengths and weakness of team members and allocating tasks accordingly.
Plan for detecting the problem	Monitoring and assessing the team leader by other team members.	Monitoring progress and speed of each group member.	Conducting research and feasibility studies.	Conducting research and feasibility studies.	Identifying strengths and weaknesses of each team member.
Mitigation plan	Change the group leader.	Conduct open discussions and further allocate tasks to assist slower group members.	Have a backup project in cases the legal risks imposed are too big.	Changing the scope of the project as a last resort.	Broadening our coding skills on our own; checking the internet and also by consulting senior students.

TEAM STRUCTURE

❖ Team Structure

NAME	ROLES	RESPONSIBILITIES
Lukundo Kampeshi	Alternative group leader	Monitors the group leader and is responsible for setting schedules, allocating tasks and monitoring completion of tasks when group leader is unavailable.
Kelvin Asare	Group member	Responsible for ensuring the team is on schedule.
Margaret Mwewa	Group leader	Responsible for setting schedules, allocating tasks and monitoring completion of tasks
Raymond Kuree	Group member	Responsible for establishing effective communication channels.

❖ Milestones

1. First Milestone: Project Proposal

The first milestone we reached was delivering our project proposal. During this process we accomplished multiple tasks such as:

- Brainstorming of project ideas: This was carried out by the four members of the group and lasted a duration of approximately one week.
- Project proposal (vision, software architecture): This was carried out by Margaret and Lukundo for a duration of one-two 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.
- Presentation: Kelvin presented the PowerPoint presentation.

2. Second Milestone: Software Requirement Specifications

The second milestone we reached was providing the software requirement specification document deliverable within one-two weeks.

- UI diagrams: The user interface diagrams were provided by Margaret.
- Product Description: The product description was provided by Lukundo within a week.
- Use cases: Margaret and Lukundo came up and wrote the use cases within one week.
- Process: Margaret and Lukundo came up with and wrote the process aspect within one week.

3. Third Milestone: Software Design Specification deliverables

The third milestone we reached was providing the software design specification document and PowerPoint presentation within two weeks.

SDS PowerPoint presentation:

- Design: Raymond designed and put together the presentation.
- Presentation: Margaret and Lukundo conducted the presentation.

UML Diagrams:

- System diagram: Lukundo, within one week.
- Class diagram: Margaret and Raymond within one week.
- Sequence diagrams: Margaret and Kelvin within one week.

SDS Document:

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

❖ **Communication**

The main communication channel is a WeChat group. We chose this channel because it is highly convenient and offers real time communication through messages, document sharing and even voice and video calls. This has helped us tackle the problem of conflicting personal schedules.

Apart from WeChat we meet during scheduled classes and occasionally during our free time.

CODING STYLE GUIDELINES

❖ HTML and CSS guidelines

<https://google.github.io/styleguide/htmlcssguide.html#General>

❖ PHP guidelines

<https://gist.github.com/ryansechrest/8138375>

❖ JavaScript guidelines

<https://google.github.io/styleguide/jsguide.html>

❖ SQL guidelines

<https://www.sqlstyle.guide/>

In an attempt to enforce these guidelines, we would configure settings in our IDE in accordance with the coding style guidelines then import identical settings in our respective IDEs.