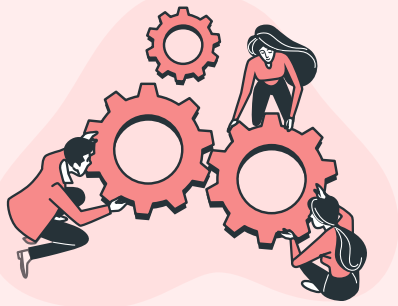


ZB

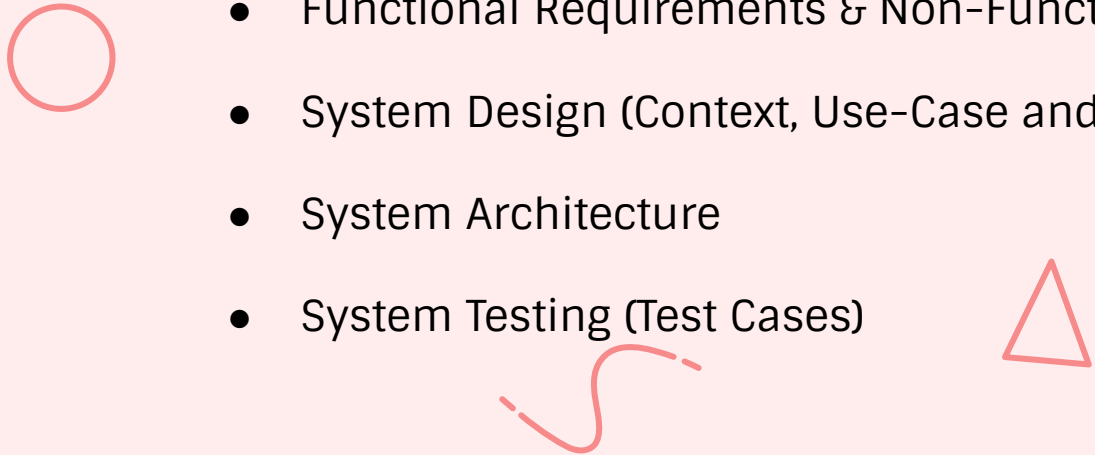
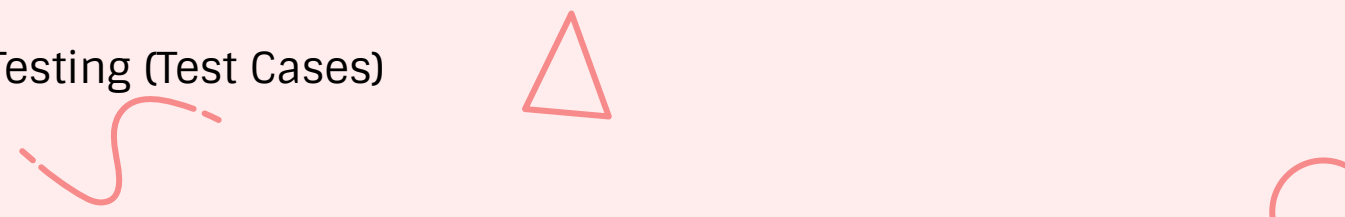
CSE2101: Submission 5

Design and Implementation of a RDBMS
for Improved Accountability





Order of Presentation

- 
- Problem & Proposed Solution
 - Feasibility Assessment
 - Risk Assessment
 - Functional Requirements & Non-Functional Requirements
 - System Design (Context, Use-Case and Class Diagrams)
 - System Architecture
 - System Testing (Test Cases)
- 

Problem & Proposed Solution

- The company John Fernandes Limited has been suffering from a lack of accountability among its employees as it relates to equipment usage and active work hours.
- To combat this dilemma the project team have found the application of a Relational Database Management (RDBMS) system to be most apt.



Feasibility Assessment

- The purpose of launching this new system is to have a record of accountability while ensuring individual responsibility and also monitoring the workflow of employees with the use of company equipment.
- Having a system that monitors these factors effectively and efficiently provides a way for employers to have logs of both equipment and employees in any potential cases of misuse or unknown responsibility.
- And provides Employees to be held liable for active work time, under company hours.
- However, introducing such a system into the environment comes varying levels of risks.

Risk Identification

Some risks identified include:

- Estimation - The projected timeline is possibly underestimated due to the unprecedented nature of the project in the domain of Guyana and the inexperience of the development team with handling this kind of system.
- Organizational - The employees of the organization may reject or under utilize the system due to it seeming overly complicated and unnecessary.
- People/Team - The development team is inexperienced with this particular type of system. It is not one that has been knowingly implemented and well documented in Guyana's context. While in theory, the staff possess the knowledge base for the creation of such a system, lack of experience can prove to be detrimental to its design and implementation.

Risk Analysis

The overall risk analysis is as follows:

Risk No.	Probability	Effect
1	Moderate	Serious
2	High	Tolerable
3	High	Tolerable
4	High	Tolerable
5	Very High	Serious
6	High	Serious
7	High	Serious
8	Moderate	Catastrophic
9	Moderate	Serious

Risk Planning

- Various risk handling strategies will be employed varying between avoiding the risk entirely and mitigation the risk entirely
- For example, for risk, number 8, the only catastrophic risk:
 - The number of hardware technologies required to implement the system may be underestimated. For example, as the company continues to expand and the number of equipment increases, the initial number of RFID tags that the system was created to process and accurately track might expand beyond the hardware's capacity.
- A suitable strategy to prevent this from happening was:
 - Additional hardware can be bought and stockpiled. In addition, a document that estimates and details the total size capacity that the system can handle can be created. This document will give the business a hard cap on the degree to which they can expand while the system remains functional and within expectations. Any further, and critical as well as costly system adjustments will have to be made.

Functional & Non-Functional Requirements

	Functional Requirements	Non-Functional Requirements
Log and record equipment condition	X	
Calculate active work hours	X	
User Login	X	
Database Query Operations	X	
Unique Identifier Assignment	X	
User		X
Administrator		X

Functional Requirements

Functional Requirements of the 'User login' feature:

- Signal transmission and receive functionality must be online.
- The algorithm must check for duplicate entries when creating random strings for the UID.
- In the event that an employee misplaces their ID and requests a new one, the algorithm must delete the old records and create new ones.

Functional Requirements

Functional Requirements of the 'Calculate work hours' feature:

- Signal transmission functionality must be online and the RFID tags' serial numbers must be appropriately named.
- The security cameras must be recording with the correct time format.
- The algorithm that scans the images must consult the correct source images in order to correctly evaluate the equipment state.

Functional Requirements

Functional Requirements of the 'Log and record' feature:

- Signal transmission functionality must be online and the RFID tags' serial numbers must be appropriately named.
- The security cameras must be recording with the correct time format.
- The algorithm that scans the images must consult the correct source images in order to correctly evaluate the equipment state.

Functional Requirements

Functional Requirements of the 'Database query' feature:

- The webpage/web app sign in page must be configured to grant/deny access given the right conditions.
- The database to handle queries and manipulation.
- Code that verifies admin login and gives access to the database for manipulation.

Functional Requirements

Functional Requirements of the 'UID assignment' feature:

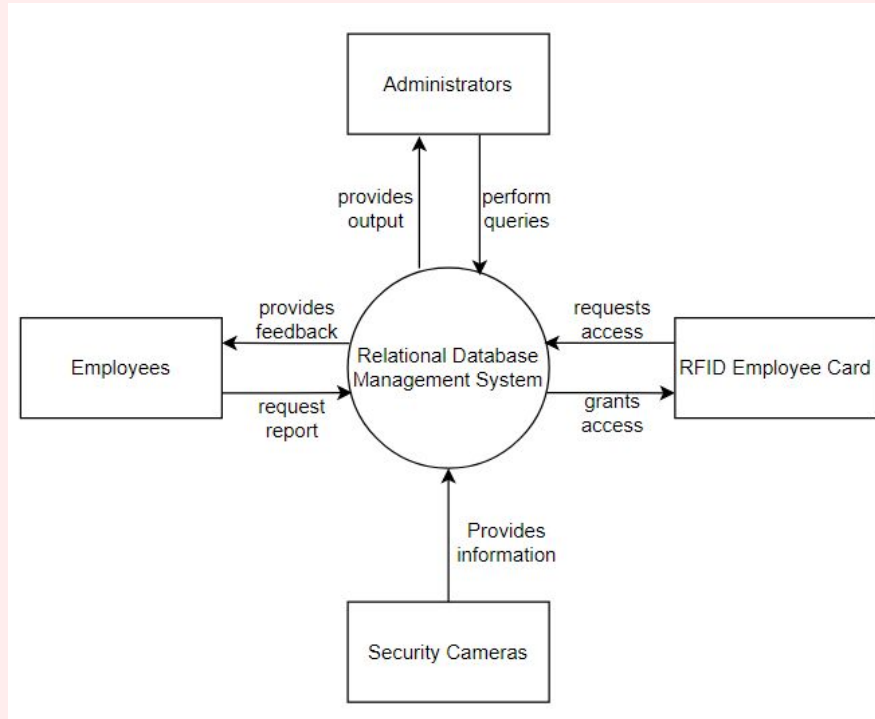
- Signal transmission and receive functionality must be online.
- The algorithm must check for duplicate entries when creating random strings for the UID.
- In the event that an employee misplaces their ID and requests a new one, the algorithm must delete the old records and create new ones.

Non-Functional Requirements

Non-Functional Requirements:

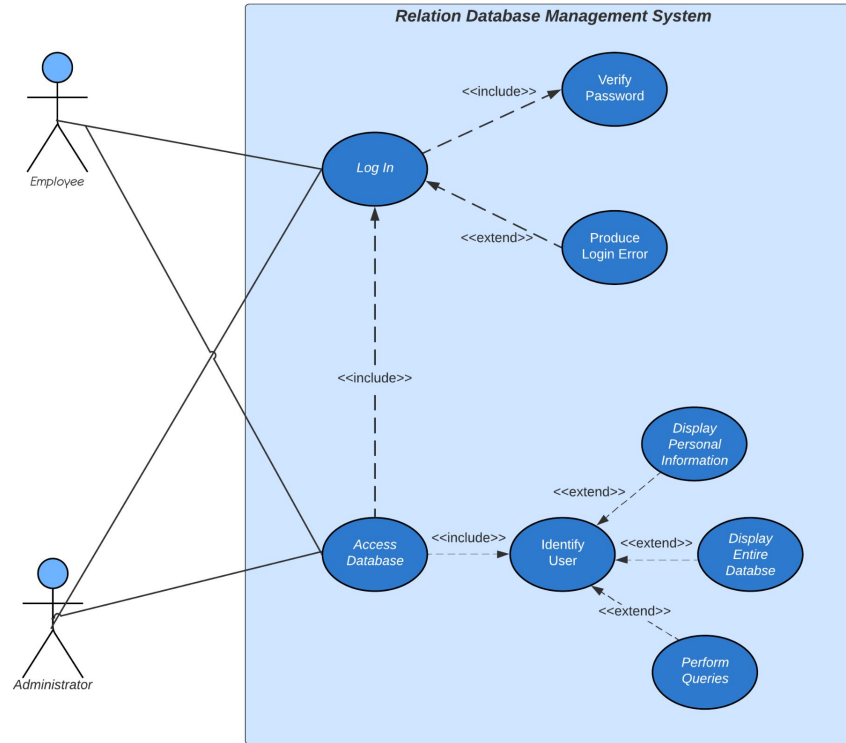
- The response rate of the RFID tags shall, on average, transmit signals to the readers once per 3 to 5 seconds (CircuitsToday, 2020). A delay in system transmission should never exceed 10 seconds. Since the purpose of these tags is to keep track of movement of work equipment, allowing for a longer delay would effectively provide an open window for malicious actions. In a worse case scenario, theft of equipment could occur.
- The database's response time for queries with small returns (no more than 20 records) should take no more than 1s.
- The database's response time for larger query returns, such as generating reports, should not exceed 30s per 1000 records.
- The system's software application should not experience lags or delays exceeding 2s at a particular time.

System Design



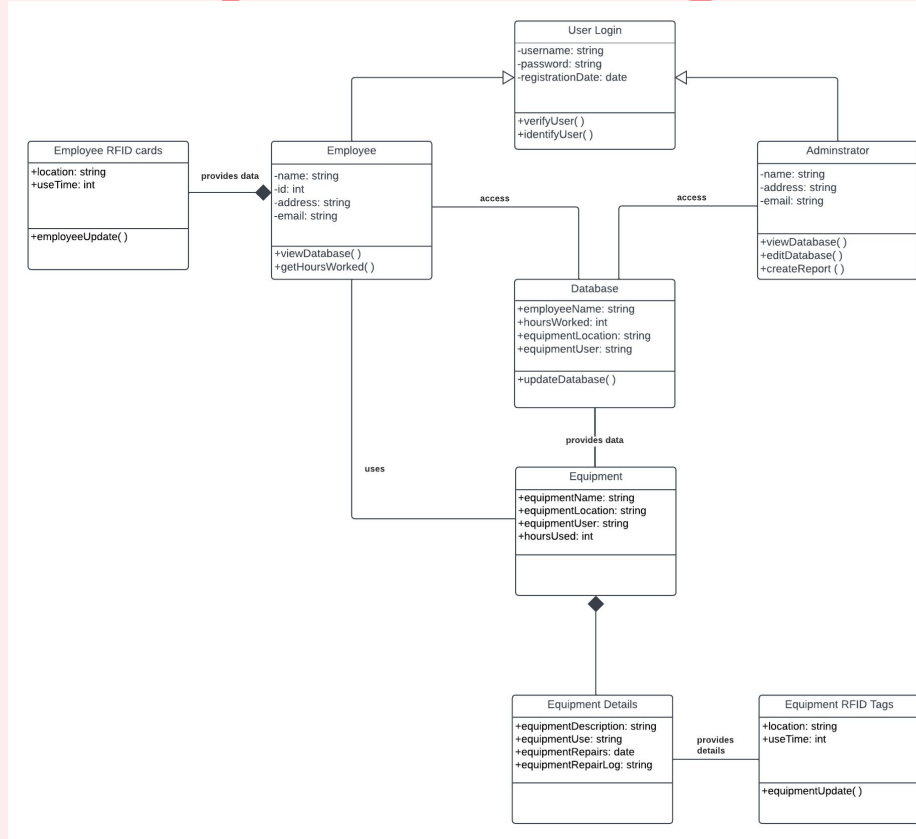
Context Diagram

System Design



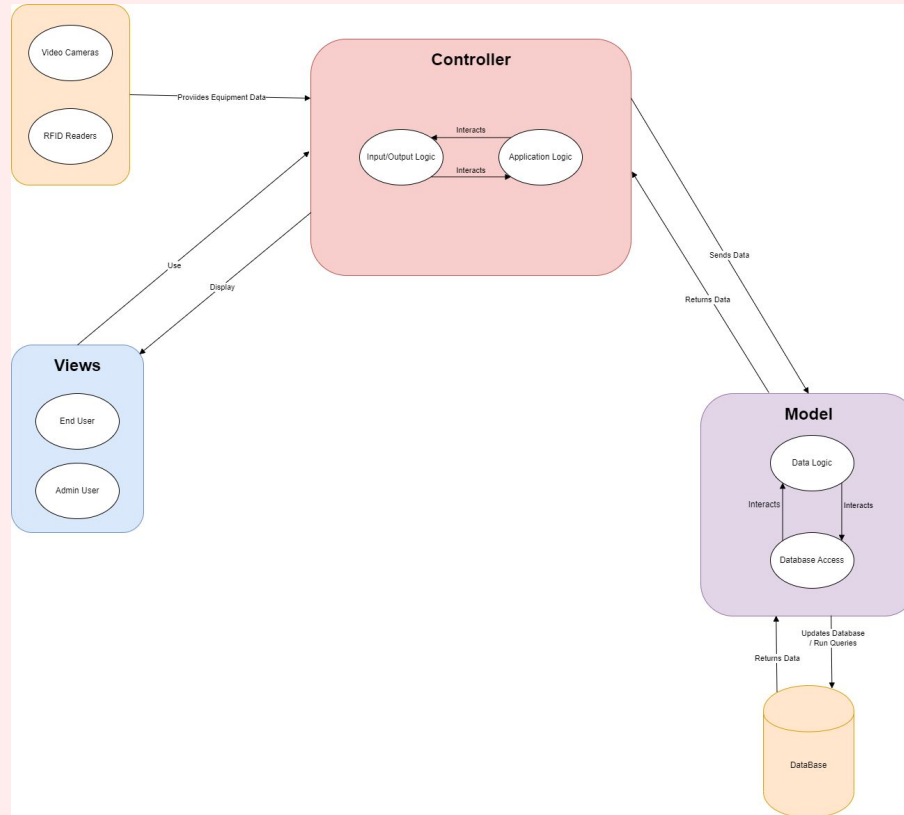
Use Case Diagram

System Design

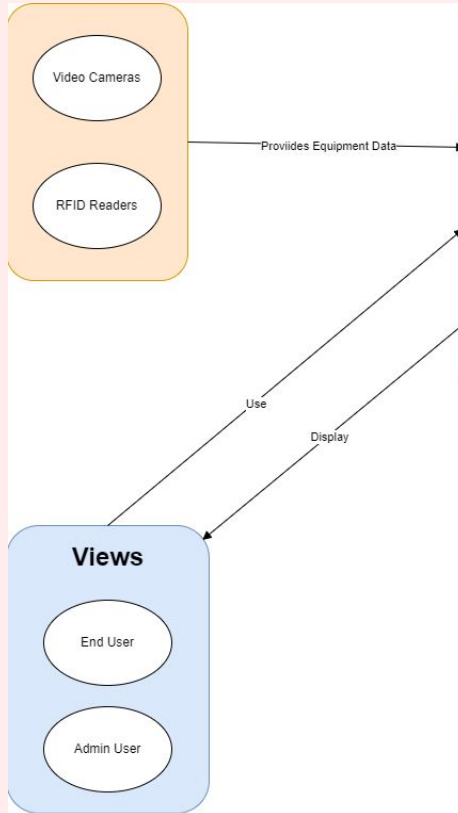


Class Diagram

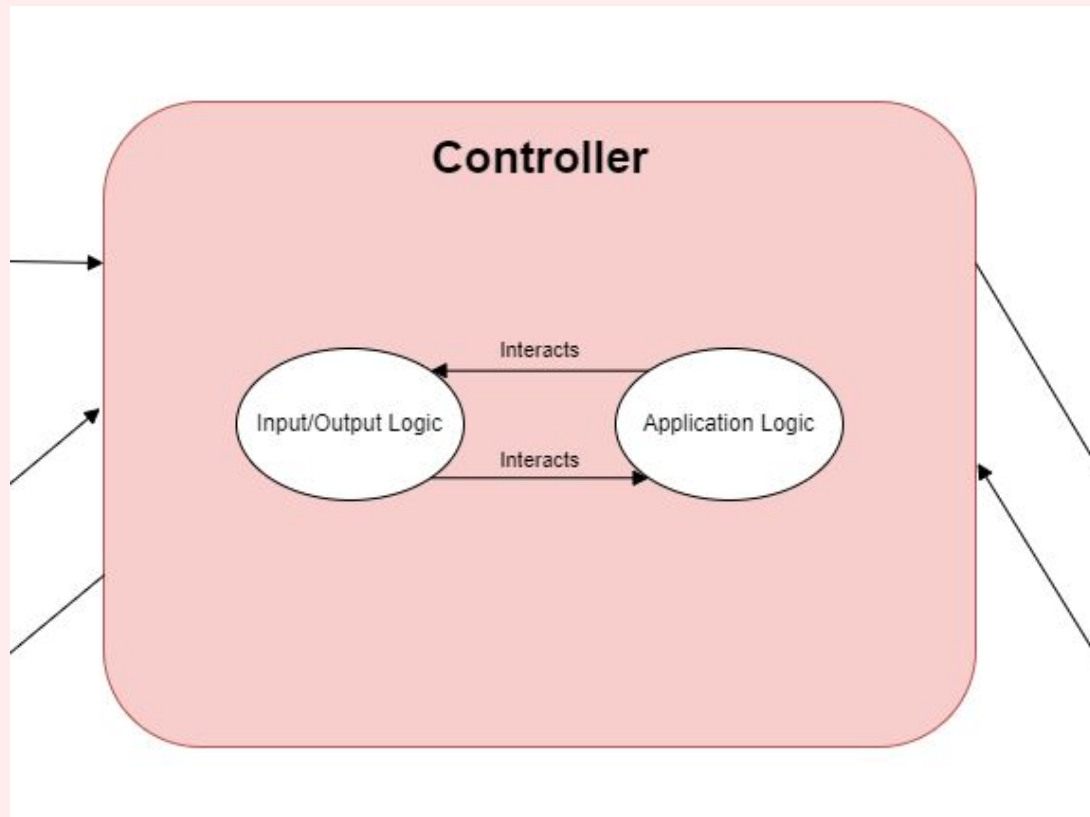
System Architecture



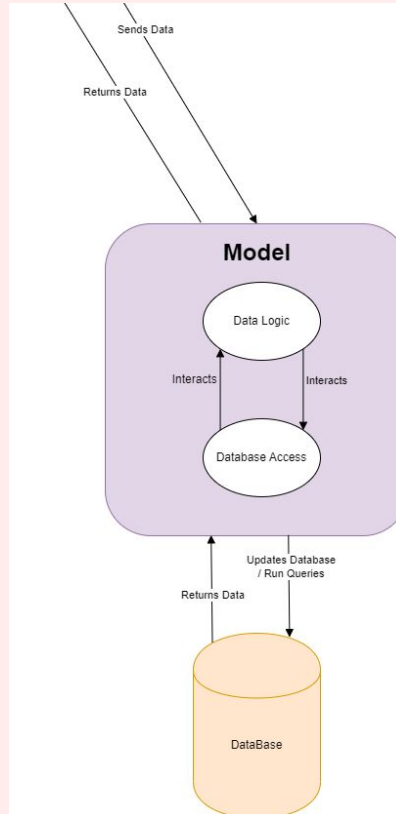
Views



Controller



Model



Test Case: Equipment condition Checking

Test case ID:	TC_UI_2
Test priority:	HIGH
Test Title/Name:	Equipment condition checking
Test Summary/Description:	This test validates the system's functionality of correctly analyzing and recording a piece of equipment's current condition.

Pre-conditions:

1. The system's database must be online and functional.
2. User interfaces components are created and functional.
3. A test setup of directly feeding the system's equipment checking algorithm must be created. In the overall system, these images would normally be inputted via security cameras but for this test case, they will be fed directly into the system via tester input.
 - a. This setup can be done by utilizing components of the user interfaces created and adding a dummy input field that collects jpegs, pngs and jpegs. That inputted image will then follow the same logic path that an image inputted via a security camera would follow.
4. Tester should have permissions to query and view the database

Dependencies:

Assume that the system has already passed the following tests:

TC_UI_1: test to see login verification is functional.

TC_UI_3: system test to check for database's functionality to appropriately store data.

TC_UI_4: system test to check that system's database responds with correct queries.

TC_UI_5: system test to check that the logic for inputting images into the system via security cameras is functional.





TC_U1_6: system test to see if admin user's interface functionality of querying and viewing the database is functional.

Test Steps:	<ol style="list-style-type: none"><li data-bbox="1000 227 1348 259">1. Open the application.<li data-bbox="1000 303 1508 336">2. Log in to the administrative view.<li data-bbox="1000 380 1574 565">3. Using the test setup interface (this is not on the admin view, it is a simple dummy field created separately), input an image of a piece of equipment in the dummy field.<li data-bbox="1000 609 1441 685">4. Wait for a response from the algorithm.
-------------	--




Test Steps

5. Using the administrative view, query the database and look for the recently added equipment condition log. This should be at the top of the database after invoking a simple select all from table name (equipmentconditionlogs) query
6. Record the algorithm's assessment of the equipment: the algorithm should report the equipment condition on a scale of: very good, good, okay, bad, very bad.
7. Record the algorithm's condition response next to the test data's assigned condition for that piece of equipment.
8. Repeat with different images from test data.

Test Data:	<p>Images: png, jpg or jpeg</p> <p>This data set will comprise pictures of onsite equipment whose conditions will be labeled as: very good, good, okay, bad, very bad.</p> <p>Admin user credentials:</p> <p>Username: 'Rosewater2fa'; Password: 'nu4]>7Pn;TZ<B;G'</p> <p>Username: 'Agelin131' Password: '!&UNCP*3HvNFW6r'</p>
Expected Results:	<p>The system should record the assessed condition of the equipment in the database table 'EquipmentConditionLogging' with the appropriate times of entry listed. Each time a piece is inserted, the administrative interface should receive an update and display the newly added log at the top of the view.</p>



Post Condition:	The data of the database's table will remain changed, without intervention, due to the insertion of the newly inputted equipment condition data.
-----------------	--



THANK YOU FOR LISTENING!
ANY QUESTIONS?

