**Group 27**

**IS 436 Section 02**

**Structured Systems Analysis and Design**

**Deliverable 5**

**12/21/2017**

Bryan Baek

Brendan Bessel

Kalabe Mulugeta

Josiah Tillett

Xinbo Yang

**Table of Contents**

1. Table of Contents…………………………………………...………….………………….1
2. Three Use Scenarios…………...…………….………….……….….…………………….2
3. Interface Structure Diagram.………………...……………………………….…..……….3
4. Interface Standards description…………………………….…..………………………....4
5. Five interface prototypes……………………………...…………………………………..5
6. Physical Data Flow Diagram (DFD) of Level 0….……………………………...………10
7. Written description of decisions made in creating Physical DFD.….………..………….11
8. Physical Entity Relation Diagram (ERD)...………….……….….….……..…………….12
9. Written description of decisions made in creating Physical ERD.….………………..….13

**Three Use Scenarios**

**Use Scenario 1: Add Employee to Records**

Manager wants to add an employee.

1. Manager authenticates into the system.
2. Manager will select **Employee Management** (1)
3. Manager will select **Add New Employee** from list of current employees. (1)
4. Manager will add employee information into the system utilizing the New Employee Form
5. Manager will confirm and save employee record. (1.1)
6. System will add employee records to Employee data store, and will make employee available in scheduling feature. (1)

**Use Scenario 2: Employee Clock-In**

Employee wants to clock-in for their shift.

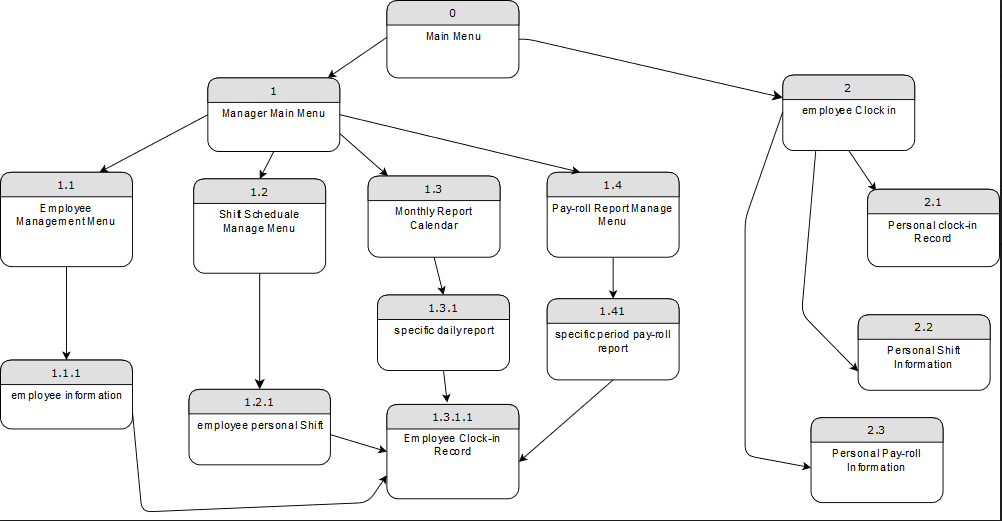
1. Employee authenticates into the system.
2. Employee will select **Clock-In** from.
3. Employee will input their employee and will verify using biometric scanner.
4. System will confirm to the employee that they have clocked in and will create a **Time-Recording Event**.

**Use Scenario 3: Checking Pay Period Report**

Manager wants to check the weekly report for discrepancies

1. System produces automated report, sends to manager for verification.
2. Manager authenticates with system.
3. Manager selects **Check Period Report**
4. Manager looks for incorrect records or notices of lateness throughout the pay period.
5. Manager makes necessary corrections.
6. Manager confirms pay period report.
7. System sends pay period report to third-party payroll company.

**Interface Structure Diagram**

****

**Interface Standards description**

* **Interface Metaphor:**

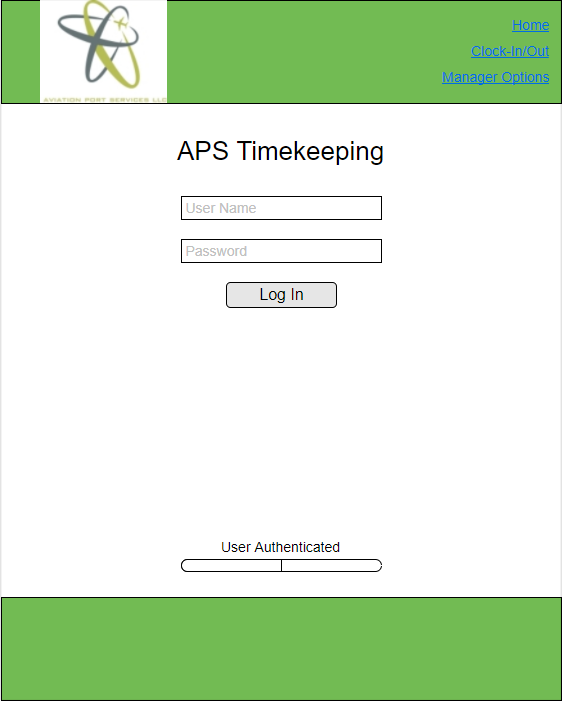
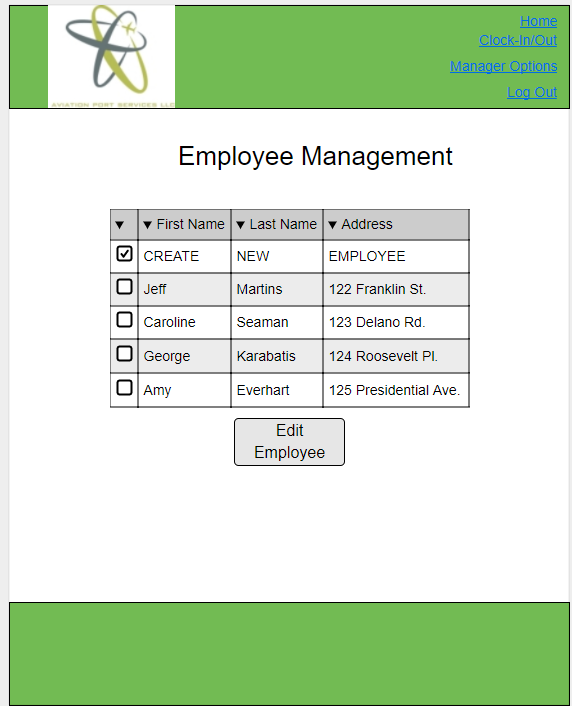
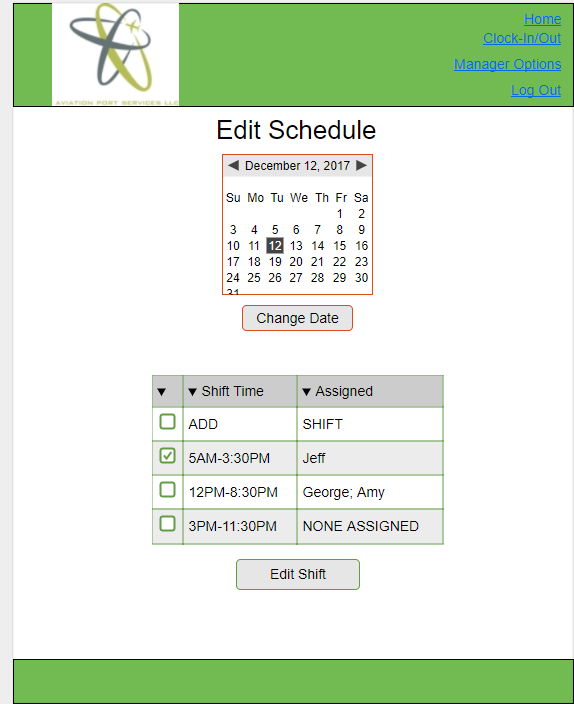
The interface Metaphor is simply generated from popular office software, T-sheets is a mature product, which fit to prediction of majority people, help user to learning fast and easy to use. We will do some adjustment, but it requires our user group to participate in test and give existed system UI as example.

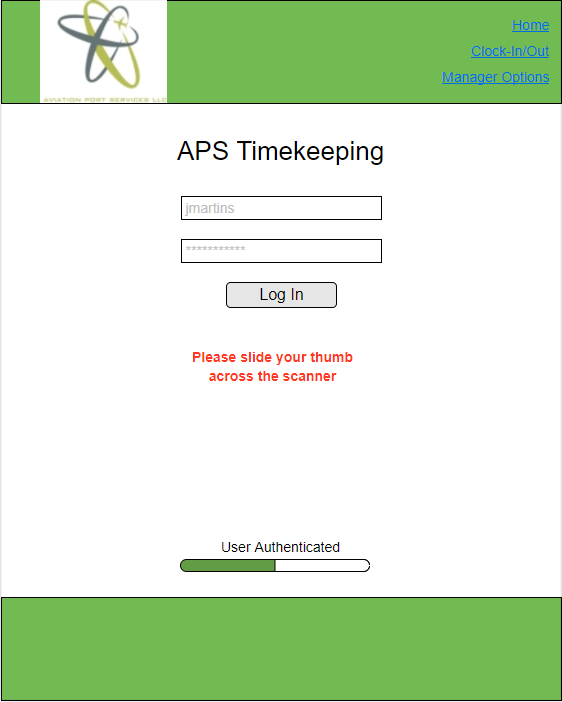
* **Interface Objects and Actions:**

In terminology, we try to follow the ‘common sense’ as much as possible. since we have variety user group, from long experience Manager, to new graduate employee, it's better to find the way that acceptable for majority rather than highly favored by minority(as long as it does not against company culture and policy). Word comment on Icon will exactly describe its function within 3 words such as “Add new Employee” or “Confirm report”. The actions will follow the design of popular office software, and try to integrate the advantage of existed system, especially for Manager. The suggestion from our user will be much valuable for further development.

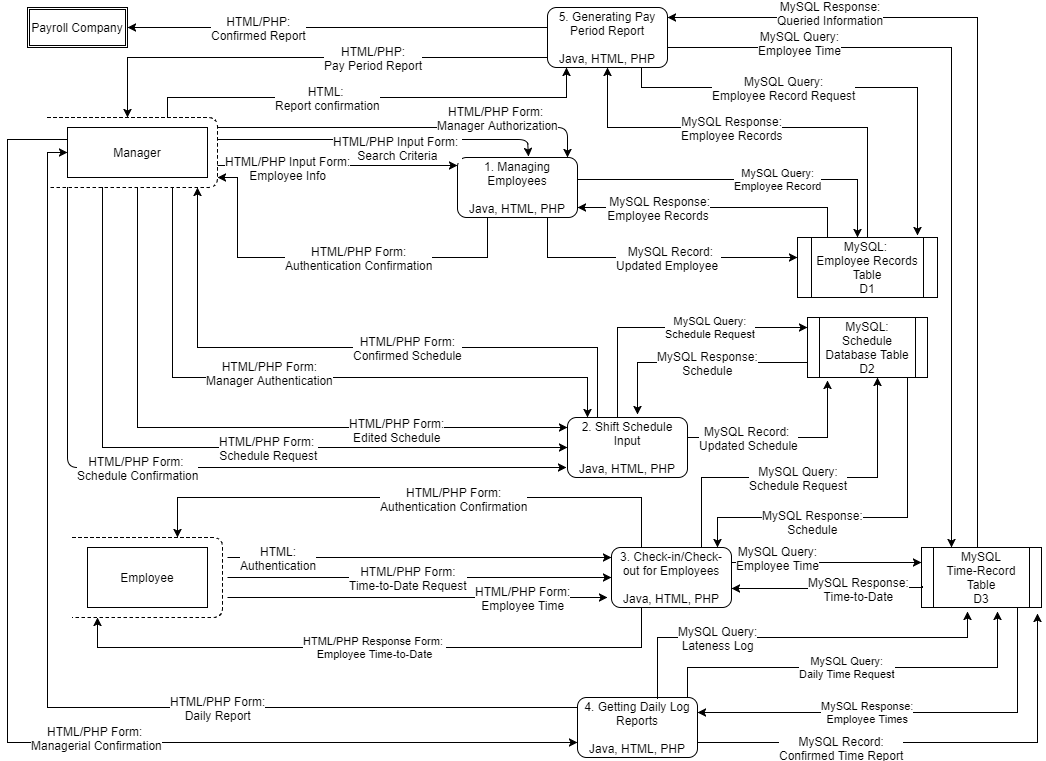
* **Interface Icons:**

We try to use commented icon as much as possible, since it provide helpful information, but comment need be limited within 3 words, especially for action like “Confirm”.

**Five interface prototypes**

****

**Physical DFD of Level 0**

****

**Written description of decisions made in creating Physical DFD**

**1.** **Add Implementation References**

Because the company we chose declined to disclose exactly how their product worked, I had to make an educated guess based on the software I was familiar with from the IS448 web development class. All data stores were implemented as MySQL databases. The processes were implemented using a combination of PHP, HTML and Java, and the data flows were implemented using labels that described whether they were HTML/PHP forms, HTML/PHP response forms, a MySQL Query, a MySQL Response, or a MySQL Record.

**2.** **Draw a Human – Machine Boundary**

For the second step, the Human-Machine boundary was implemented around the Manager and the Employee because those were the only two users of the system.

**3.** **Add System-Related Data Stores, Data Flows, and Processes**

For this step nothing needed to be added because we were using pre-packaged software. I felt that the system was fine as is and chose not to add anything to alter how the system worked.

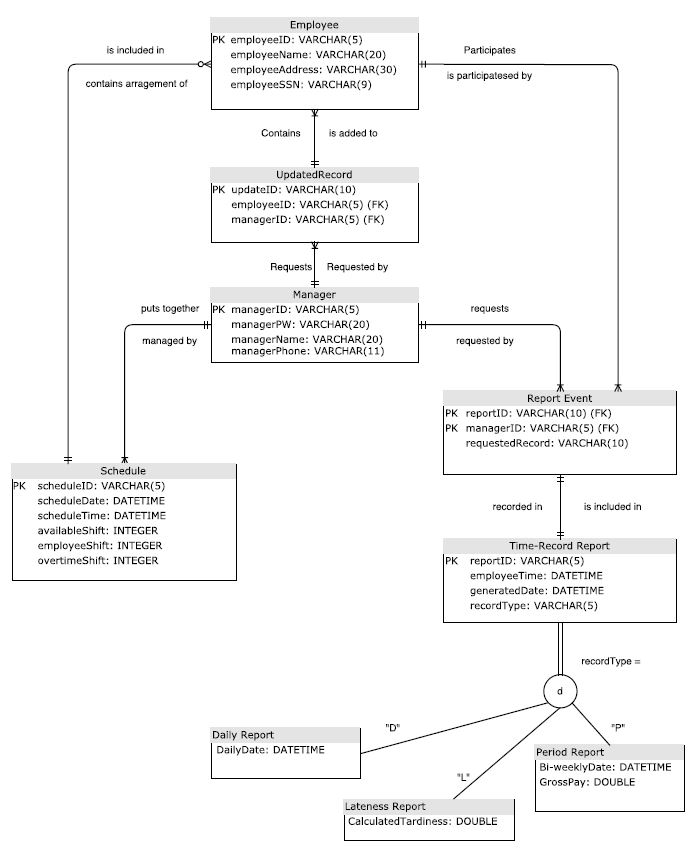
**4.** **Update the Data Elements in the Data Flow**

Because we used prepackaged software this step was non-applicable.

**5.** **Update the Metadata in the Computer-Aided Software Engineering Repository**

This step was also non-applicable because we didn’t use a CASE repository in our project.

**Physical ERD**

****

**Written descriptions of decisions made in creating Physical ERD**

1. **Adding Foreign Keys**

We added foreign keys to make the relationships of the tables more apparent to each other. We did this by identifying the necessary relationships between tables, and then adding the primary key as a foreign key to the other table.

1. **Adding Data Types**

We added data types in order to make the specific function of each column more apparent to the team who would be developing the database to go along with the system. For our primary keys, we ended up choosing a string variable so we wouldn’t be bound to using a number as our employee ID, making the system just a little more secure.