King Fahd University of Petroleum and Minerals
Computer Engineering Department

Senior Design (COE485)
Semester 141

Electronic Parts Tracker
(E-Keeper)

Design Document

Team Member:

- Wael Elagi
- Mahdi Al Mayuf
- Abdulrahman Alkhiary

Advisor:

Dr. Ahmad Khayyat

20/10/2014

Table of contents

List of tables

List of figures

1 Introduction

This project was proposed by the Computer Engineering Department (COE Dept.) to improve the department resources management. The COE Dept. has a lot of resources, but the scope of the project is to deal with the electronic parts and devices. Tracking these parts is done manually in the current procedure which makes it hard and sometimes impossible to track, account for parts, and creating reports of the inventory. This project is about developing an information system that keeps records for all the parts in the COE Dept., and track them using a data entry mechanism to be used by multiple types of users.

2 Problem statement

The existing procedure has many drawbacks, it consist of paper logs and Excel sheets, which makes it hard to keep track of the parts going through a huge amount of paper and sheets. In addition, the forms are not unified, there are parts that are not tagged to be distinguished from other parts see Figure 1.

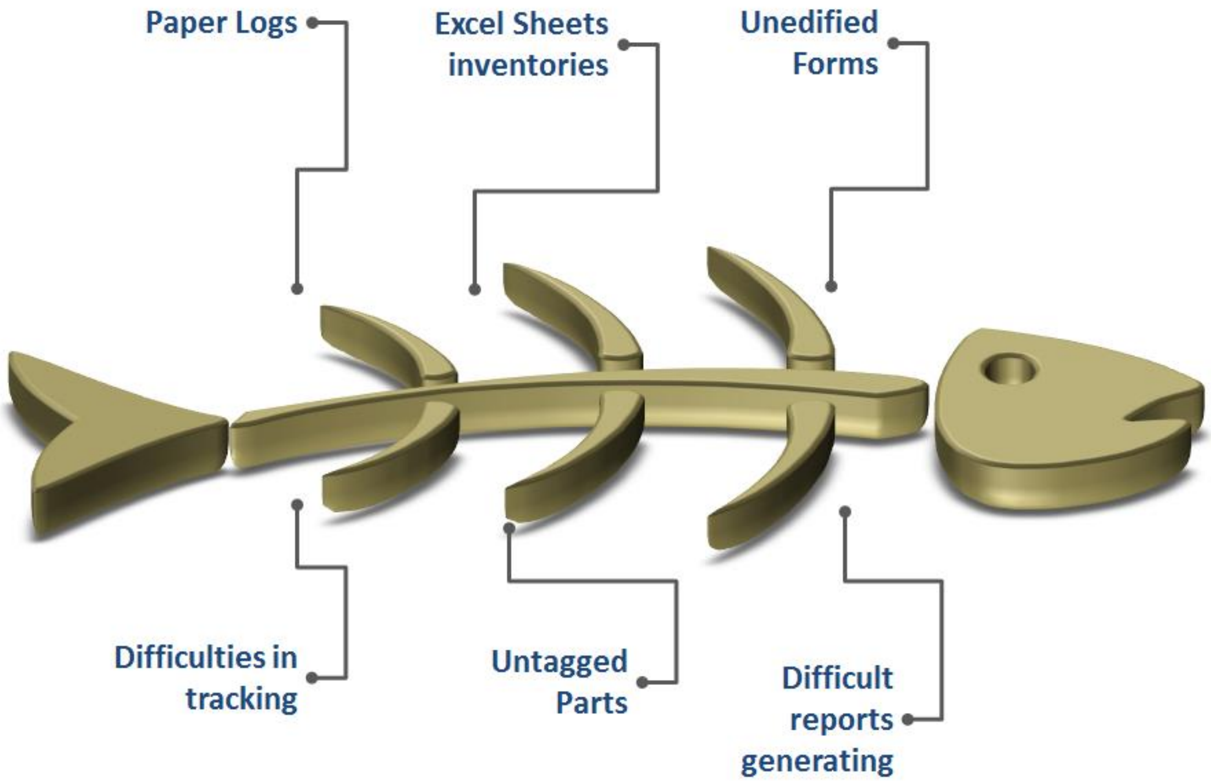

Figure 1: Drawbacks Fishbone Analysis

The project is a system for tracking electronic parts of the COE department, and managing part borrowing through approving process. It should allow its users to check the availability of a given part and to borrow it. A part is available if the department has it and it is not checked out (borrowed).

2.1 Project impact

1. Saving time, efforts and recourses.
2. Accessibility.
3. Automation of the request/approve process (Less paper work).

3 Project Specifications

This section is listed the project requirements and how these requirements meet the specifications.

3.1 Requirements

- Add new parts and their quantities. Similar but non-identical parts should be distinguishable. Include part photos.
- Flexible categorization of parts
- Allow students to request parts, the request should be approved by faculty and the part owner.
- Allow faculties to request parts, the request should be approved by the part owner.
- Allow the part owner to approve the request.
- Only the Admin can change the privileges for each user.
- Keep tracking for request date, borrowed date, due date and returned date.
- The system support history for each part borrowed.
- Track individual parts to show whether a part is available or borrowed.
- Filter parts by category.
- Configurable privileges.
- Deployment of the system in a way accessible to all intended users.

3.2 Specification

- The system is built as a web application system to be accessible for all the users.
- A database that keep records of parts and their quantities, users, requests and approvals
- The ability to generate reports about parts history and status.
- Catalog to view the parts availability and status.
- Generating a unique QR code for each part.
- Using a scan gun to read the QR code and serial code.
- Using a Camera to capture photo for each part.
- The system will provide a different views for different user depends on their privileges.

4 System design

4.1 Architecture

The system consists of two main components, Server and Client. The Client is basically a web Browser, and the server has three primary components which are, web server to serve HTTP requests, a web application server to serve the business logic, and a database. The web application server includes three components:

1. User management: user types, privileges.
2. Request management: request generating and approving
3. Part management: Part information, status and accountability.
4. Request Handler: Handle the Business logic requests, and invoke the system components.

   See Figure 2:

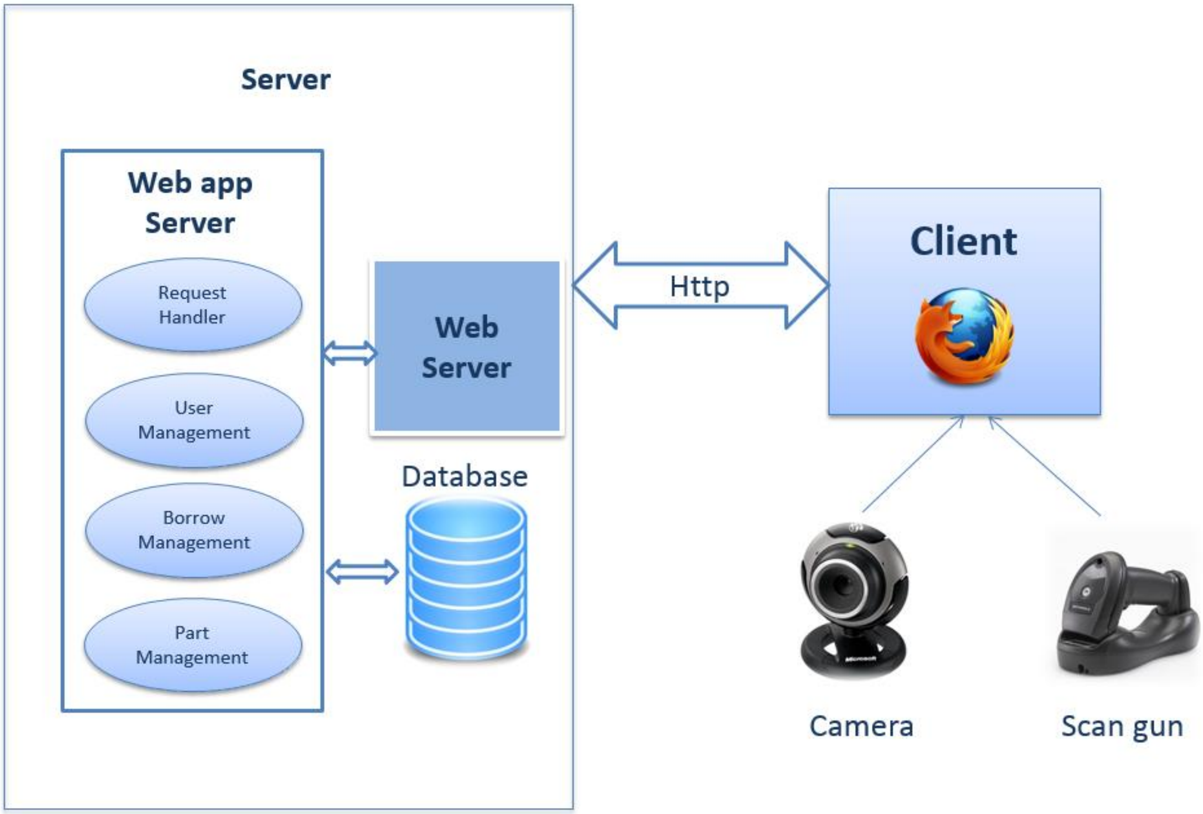

Figure 2: System Architecture

4.2 Component Design and Implementation

4.2.1 Components

The system has two types of components, readymade and custom. The readymade has hardware components and software components. Which are shown in Table 1:

Table 1: Off-shelf Components

Hardware components	software components
• **Server (workstation)** • **Scan gun** • **Cam**	• Web Application server • Web server • Database server • User Management (Through Framework) • Client (web browser)

The custom component is all software, so they need to be implemented to meet the system custom specification. These components are:

- Database
- Part Management
- Request Management
- Request Handler

4.2.1.1 Other Software Components

The other software components are going to be implemented Using Play Framework, and they are designed as followed:

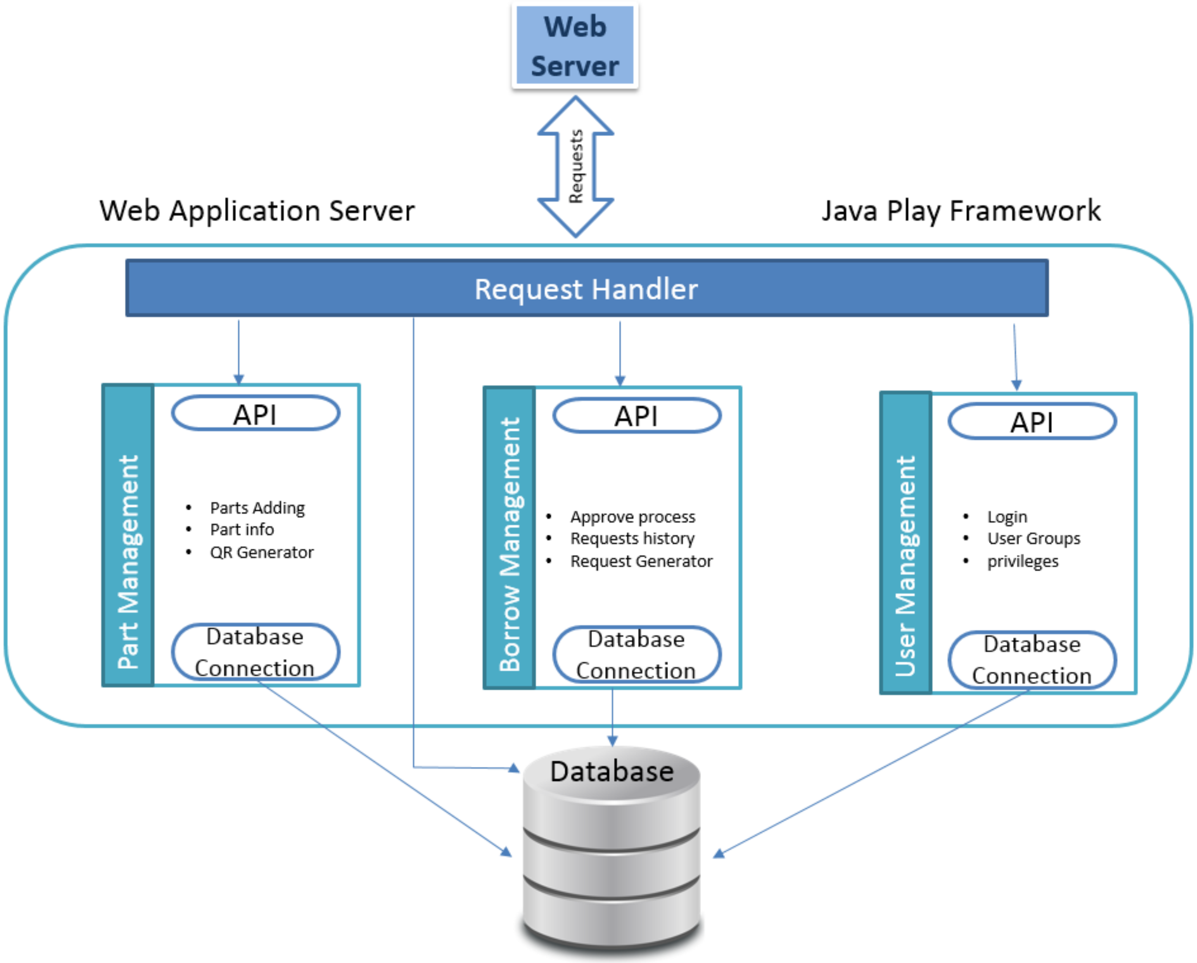

Figure 3: Web Application Design

There are four main components which are User Management, Borrow Management, Part Management and Request Handler. Each of these components has an API that specifies their operations and database connection to store and retrieve data. The above figure also shows the operations performed by each component. The request handler will receive the request and invoke the concerned components.

4.2.2 Design

The web application is going to be implemented using java with Play framework. There are also some hardware components such as scan gun which is used to scan QR codes and a cam to capture a photo of the part when it is first added. These two components are connected in the client side (web browser).

4.2.2.1 Database design

The database schema is shown in Figure 4:

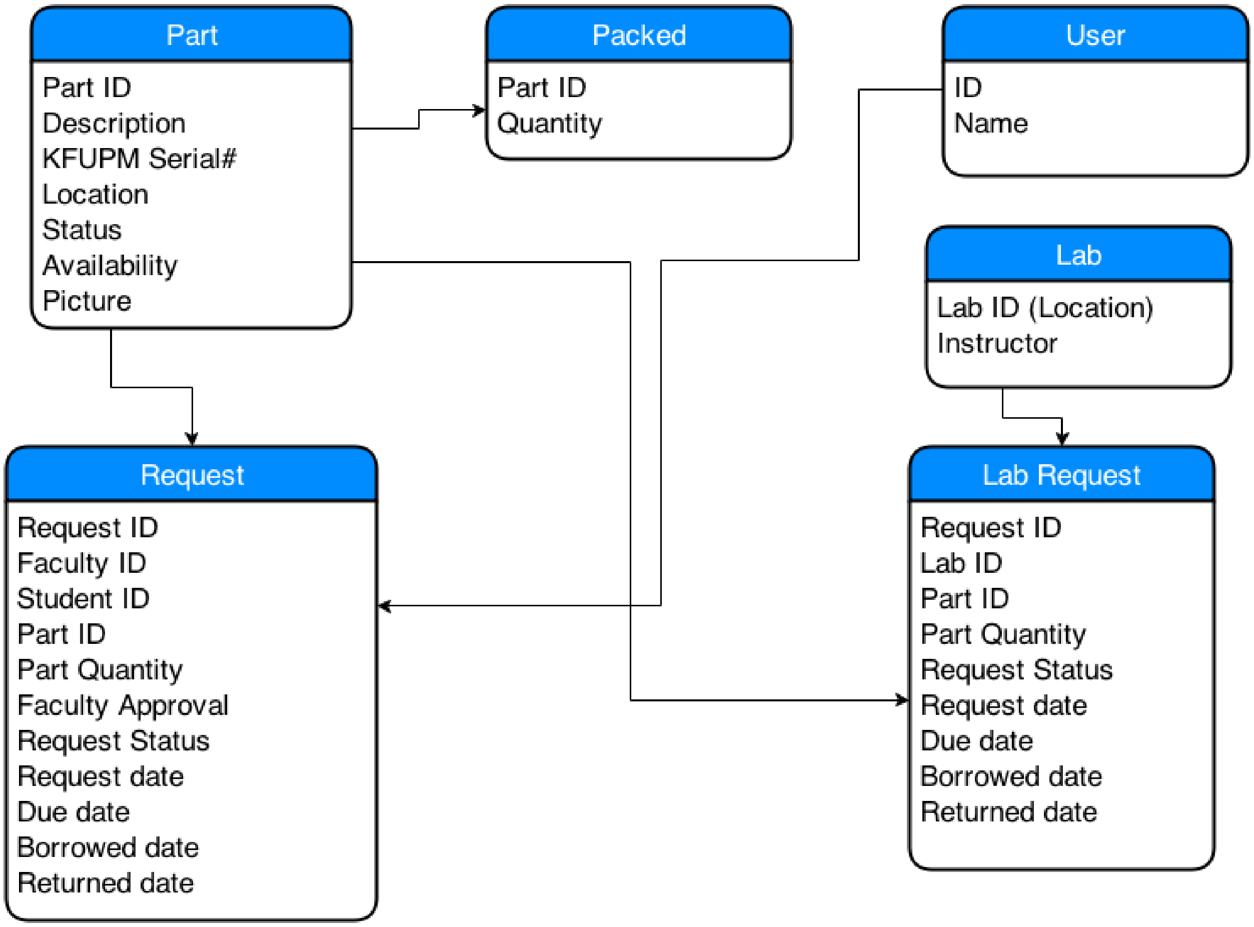

Figure 4: Database ERD

4.3 System Integration

For the main components which are the client and the server, they interact using HTTP requests. Software components are interfaced using their API, and they interact through the database and the request handler. Finally, hardware components like the scan gun, it interact with OS through USB ports and the web browser (client) take the readings form the OS.

4.4 Design Decisions

There are two options to implement the system. The options are web application and Desktop application. The following table shows the tradeoffs between the two options.

Table 2: Platforms Tradeoffs

Options Criteria	web application	Desktop application
Accessibility	Anywhere	local device
Development	One system(multiple views)	multiple system
Installation	One side	Many sides
Component integration	One environment	Many environment
System Update	Server side	All workstations
OS independence	No	yes
Server Load	High	Medium

The following table compare the options depends on the criteria's weights. The comparison scale used from 10 – 0 (10 Max, 0 Min)

Table 3: Scaled Comparison

Options Criteria	Weight	web application	Desktop application
Accessibility	0.3	10	7
Development	0.2	7	5
Installation	0.15	8	5
Component integration	0.05	7	6
System Update	0.1	8	5
OS independence	0.15	9	1
Server Load	0.05	4	**7**

The following table shows the final result.

Table 4: Comparison Result

Options Criteria	web application	Desktop application
Accessibility	3	2.1
Development	1.4	1
Installation	1.2	0.75
Component integration	0.35	0.3
System Update	0.8	0.5
OS independence	1.35	0.15
Server Load	0.2	0.35
Result	8.3	5.15

From the previous tables, the web application has more features than desktop application. As a result, the project will be implemented as a web application.

4.4.1 RDBMS VS ORDM

The Relational Database Management System (RDBMS) is a database that is based on relational model that stores data in the form of related tables. On the other hand, the Object Relational Database Mapping (ORDM) is a programing technique that utilized through an Application Programming Interface (API).

ORDM is easier in development side than RDBMS which allows developers to convert data from rich data types used in object oriented programming languages to lower level relational database types. Also, ORDM has high efficient than RDMBS in the small/medium applications that aren’t accessing the database frequently. The database schema will be implemented using ORDM.

5 Progress

The following tables will show the project tasks along with their description and duration, as well as the team accomplished tasks so far. See

Table 5 and Table 6:

Table 5: Tasks

	Tasks	Owner	Description	Duration (weeks)
1.0	**Requirements and specifications**	Khiary		
1.1	Analyzing Requirement	Mahdi	Analyzing the requirement and finalizing the functional and nonfunctional requirements	1
1.2	Specifying the specification	Wael	Specifying the specification that meet all the requirements	1
2.0	**Database design**	Wael		
2.1	Logical design	Wael	Design Entity Relation Diagram (ER)	1
2.2	Physical design	Mahdi	Implementing Logical design in the database server. This task has been delayed to the framework development section (task 4.2) to implement it using Object Relational Mapper (ORB)	-
3.0	**Platform and Framework**	Khiary		
3.1	Evaluating and choosing platform	Mahdi	Evaluating platform options depends on specific criteria	1
3.2	Choosing framework	Khiary	Evaluating the available frameworks	1
4.0	**Developing Web application**	Wael		
4.1	Setup developing environment	All team	Installing the framework (Play) and other components	1
4.2	Implementing database	Mahdi	Implementing database using Object Relational Mapper (ORM)	1
4.3	Developing Part Management	Wael	Generating QR code, developing Add part function and integrating hardware components (Scan gun, Cam)	2
4.4	Developing Catalog	Khiary	Catalog that give the ability for the user to view the parts in the system	2
4.5	Developing User Management	Mahdi	Design Login function, User group and privileges	2
4.6	Developing Borrow Management	Khiary	Generating part request, approval process and Request history	3
4.7	Integrating all components	Wael	Collecting the different components and built the connection between them	1
5.0	**System Deployment**	Khiary	Deploy the system in the server and its accessibility	1
6.0	**System Testing**	Mahdi	Check the system functionality and fix any errors.	2

Table 6: Timespan

Weeks

	Tasks	**Status**	3	4	5	6	7	8	9	10	11	12	13	14	15
1.0	**Requirements and specifications**	Completed	X	X											
1.1	Analyzing Requirement		X												
1.2	Specifying the specification			X											
2.0	**Database design**	In progress			X										
2.1	Logical design	Completed			X										
2.2	Physical design	Delayed			-										
3.0	**Platform and Framework**	Completed				X	X								
3.1	Evaluating and choosing platform					X									
3.2	Choosing framework						X								
4.0	**Developing Web application**							X	X	X	X	X	X		
4.1	Setup developing environment	In progress						X							
4.2	Implementing database	Waiting							X						
4.3	Developing Part Management	Waiting								X	X				
4.4	Developing Catalog	Waiting									X	X			
4.5	Developing User Management	Waiting									X	X			
4.6	Developing Borrow Management	Waiting									X	X	X		
4.7	Integrating all components	Waiting											X		
5.0	**System Deployment**	Waiting												X	
6.0	**System Testing**	Waiting												X	X

6 Completed Task

- **Requirements and specifications**
    - In the beginning, the requirements were really cleared until we met with Khaled Malek. After that, we analyzed the functional and nonfunctional requirements. Finally, specified the specification needed to meet the requirements.
    - **Problem faced:** We didn't know how the current system is functioning until we met with Khaled Malek.

- **Database design**
    - After finalizing the requirements and specifications, we designed the database starting with Entity Relation diagram (ER).
    - **Problem faced:** Identifying the entity's fields and we solved this issue by reviewing the old forms provided by Khaled Malek.

- **Platform**
    - We had two options in the platform which were designing the system using Web Application or Desktop Application. We compared these options depends on specific criteria.
    - **Problem face:** No problem faced.
- **Framework**
    - We consulted with Dr. Ahmad Khayyat on the available web application frameworks and he suggested using Play framework or Django framework. The team members chose to develop the system using Play framework since the team members are familiar with Java.
    - **Problem faced:** We didn't have previous knowledge in web application frameworks and we solved this problem through consulting and researching.