A business often needs to purchase items from external vendors. This process can easily become disorganized, especially in large companies where employees can lose time asking each other about orders and sometimes not even realize that the same order has already been requested. So, here we come to solve this problem with Order Processing Application solution. The project is to develop a web based Order Processing application to keep track of the complete list of items ordered by the Client company. The purpose of the application is to give coordination between departments (and/or employees) so the company does not order more items than necessary. This way, large companies can avoid losing time and money from inefficiency and overspending. Thus, with this application project, our goal is that employees from across the company can see: what has been ordered? who ordered it? for which project/department? how many? where? what is the cost? how long will it take to arrive? In addition to these, The Order Processing app would also useful in:

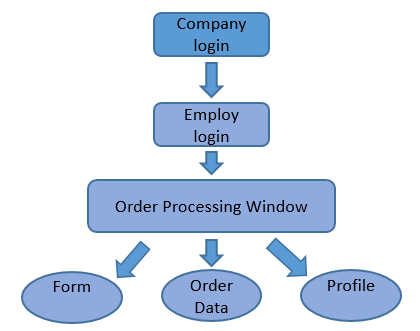
* Make it easy to calculate the expenses of each project or department for marketing and accounting team
* Save time by automatically notifying a project manager when a new order needs approval or notifying the department when an order has arrived
* Centralize ordering so the company doesn't need to provide financial account information to every department

A proprietary software systems does exist for order management sold by a variety of companies. After research and study, we have found that some application developing companies like Exence, a Polish software company, has a product that happens to be very similar to ours, except it is a desktop application. A prominent problem with a desktop application is that user is limited to their work network and cannot access the application from home or other places. In this developed software industry, everyone wants quick and almost instant gratification. Therefore, we planned to make a web based application which slightly mimics their software. Moreover, users can access from home or other places with other devices. Ours is a web app for ease of access and adding a desktop client for our application could be a potential extension of the project in the future.

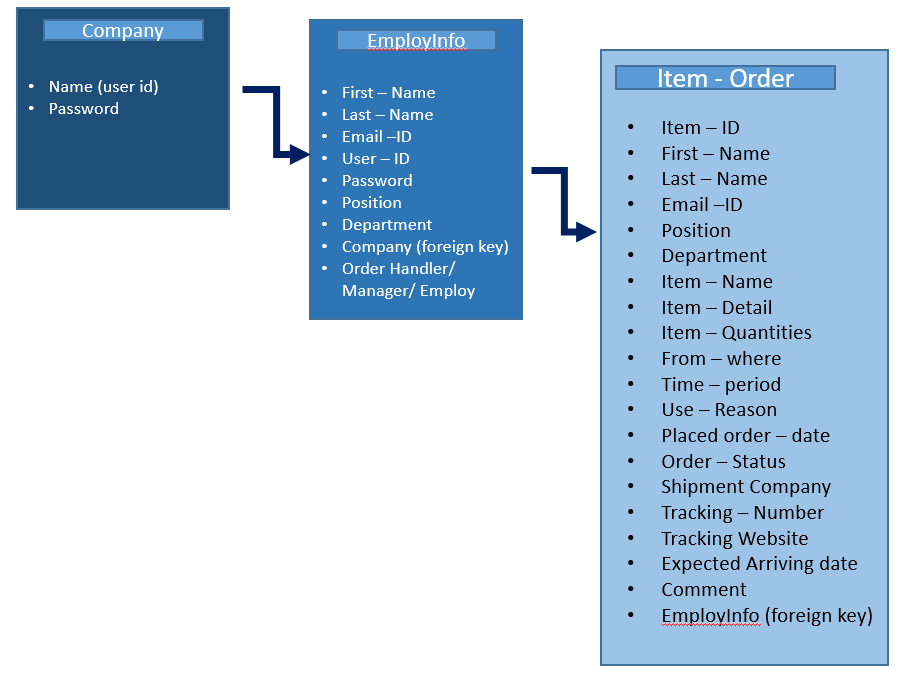
Our app will run in a web browser and consists of a user interface and a back-end database. The application works in two parts: (1) Requesting new order by completing a digital order form (2) Table which contains a list of all the orders that exist within the company. In addition to checking and keep tracking of orders, users who have managerial privileges and are order handlers, can confirm the orders and notify the user who requested the order. Order handlers can add order details such as the tracking number, tracking website, shipment detail, order placed date, and expected arriving date. When an order arrives in shipment, order handlers can change the “placed order” status to “completed order” status and notify the person who requested the order.

Regarding the development work, it consists of front end, which is creating graphical user interface and a back-end which is responsible for updating data within the database and handling routes. The client-side is programmed using JavaScript, HTML, and CSS. The back end is consisting of a database that is contain all user account, order objects and their attributes. Main view in UI will be populated directly from the DB. Also entries will be added, deleted, or modified based on the front end users’ actions. The back end work for the app is programmed using flask, python and JavaScript with ajax calls.

The project undertook a modified agile model. A GitHub repository was set up so that all tasks which were needed for project and fulfill the project requirements were included in the project backlog. The basic architecture design of interfaces is below. And a detailed description of all inputs into and outputs from the application is following after diagram:



The database structure diagram is shown below which is containing Data entities and their relationships:



The data for the Order Processing Application system catalog was stored using a MySQL database, and is hosted on the Google Cloud. All Group members have access to this database and permission to add or edit tables as needed for the system. Only the developers have access to delete and manipulate the tables. The users and managers will not have direct access.

Flask is a powerful python web framework which is extremely useful in creating a small server which can handle web page routing. Since it is a python framework the python libraries and other extensions are also available to make the web development process easier. Flask does not create HTML web pages, but takes HTML files that already exist as templates and renders them to the local-host. Employees and Managers are users and they will need to log on to the web page. External python libraries are used to include password encryption which will protect their passwords by encrypting them before they are entered into the database.

Tested all code from all team member work. Tested individually first, then integrate the entire Group’s work for that sprint and tested that the whole system works together. Also, tried to test each other’s code, so that way if any bug or issue was found it was resolved before delivery. Most of testing was done by running each other’s code and having the intention to break it as much as possible. If it could stand up, the program was valid and acceptable.

The libraries and tools used in this project are all free or open-source and should not present any licensing issues, and to prevent any issues we included the licenses for the authors that have provided us explicit consent. The resulting product is not intended to be sold and is not meant to compete with any other software system with the same or similar functionality. System shall give limitation for user to work. This may seem backward at first but, by understanding and implementing this constraint when designing and building system will help users navigate and use the system with minimal errors, thus creating a more satisfactory user experience. The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection needed for the application to function. This system is usable 24/7. The application is portable with any browser (preferably Chrome) and on any operating system. System was tested on different browsers and on different operating systems. The reliability that the system allows the user to login with both company and employ account information. Also system allows user to fill the form, edit the profile information, and check the order data in table. User with authorities is allowed to edit the order information and status of the order by the system. System works smoothly without any delay. System only allows editing authority to manger or order handler user. Multiple users are able to login at same time. Order data or account information is getting updated on database with in 1 second. If a user is not using the application system, the system must log out the user for security. The application should be easy to extend for maintainability. The code has been written in a way that it favors implementation of new functions.

Based on the research and the analysis of this feasibility study, the team had agreed that the project was FEASIBLE and the team was WILLING to accept the challenges of the project and will work to fulfill all its requirements. Some requirements have not been explicitly stated, or the details were not fully specified, leading to ambiguity when deciding how to proceed with implementation. The group, made sure all members were clear on what needed to be done and how it should be done before proceeding. Team members asked questions to other team members or to someone with expertise in that resource/tool and avoided wasting time on a single entity.

In the future, some possible non-functional requirement categories include: speed optimization, extra error handling, minimum number of concurrent users to support, and granting users the ability to print the order data. As mentioned before in the introduction, we have a web app for ease of access and adding a desktop client for our application could be a feature for future extension of the project. In the proposal regarding this project, the team agreed that a system shall contain a notification feature for orders, but due to the lack of time and one team member leaving the company, the team was unable to accomplish this requirement. And we are now planning to add this functionality in future work. Also email notification requires proper security so team has planned to do more research for this and achieved in future project.

Thus, the main experience we learn from this project is to work diligently in group, and manage and stay within the project timeline. The team was small because it consisted of only 2 members, and both were not technically oriented experts and they have limited knowledge of relevant web. However, both team members learned about web technologies such MySQL databases schemas, flask framework, SQLAlchemy ORM’s and Ajax calls in addition to some programming languages such as Python and JavaScript. However, as a good improvement point, we can improve given that we had enough experience and plan our requirements according to the resources that we have. Therefore, we can have a more powerful and successful project.

**Roles:**

|  |  |
| --- | --- |
| Muhammed Khalid | ScrumMaster and SW Developer |
| Vibhuti Patel | Product Owner and Principal SW Engineer |

**Ceremonies and artifacts:**

Created a product backlog which includes features that we want to add in our product. Using scrum, we created a product backlog in our first Scrum meeting. Additionally, we took features that we had put into the product backlog and added to a sprint one backlog. The features that were added to our sprint one backlog, were tackled in that sprint.

|  |
| --- |
| **Add MySQL ORM** – a tool that will allow database insertion and extraction with ease (we used SQLAlchemy for our project) |
| **Give Edit Permissions for Orders depending on User Type** – Provide different users with different privileges. Employees will able to see orders they placed while managers will be able to approve and reject orders. |
| **Create UI Showing list of orders** – A User Interface that will show list of orders that have been placed by each user. A user will be able to see an order that he has placed. |
| **Create UI Allowing user to modify profile information** – Create a separate user interface that will allow users to modify their profile information (i.e email, position, department, etc.) |
| **Create User Login Sessions** – Implement a system that will allow users to login with unique profiles for them. That way different users will be able to login and place orders and modify their own profiles. |
| **Filter displayed orders depending on type of user** – If the user is a manager, he may see all orders that have been placed within the department. A regular employee user will only have access to orders that he or she has placed |
| **Modify Login UI** – In the first sprint, we created a login that was bland and straight to the point. We want to build upon from the first sprint by improving the GUI with CSS and JavaScript. |
| **Implement password encryption when storing inside MySQL** **database** – Do not want raw passwords in the database because that is very unsafe. Encrypt the password and then store it I the database. More for user protection |
| **Add backend to User Submission UI –** This is an artifact that was rolled over from the first sprint. We were not able to complete it, but completed it this sprint. Added the functionality for users to place orders which would be appended to the database. |
| **Create New Order Submission UI** – Create a UI or GUI for the client-side. This is what the user will see when placing orders. |

**Sprint Planning Meeting**

For our sprint planning meeting, we had a product backlog of features that will be implemented in our product. We took a few items for the product backlog and decided who would do what. We took 7 items from our product backlog and planned to tackle them in our first sprint. Below is our finalized sprint2 backlog with stories, respective assignees, and an estimate for each calculated in hours. Our Sprint Planning meeting was conducted on March 28th, 2017 in Lyndon Library. All two members met

together and discussed and finalized the sprint backlog.

| **Stories** | **Assignee** | **Estimate (Hours)** | **Status** |
| --- | --- | --- | --- |
| Create UI allowing user to modify profile information | Vibhuti Patel | 2 | Open |
| Give Edit Permissions for Orders depending on User Type | Vibhuti Patel | 1 | Open |
| Add backend to User Submission UI | Muhammed Khalid | 5 | Open |
| Create UI showing list of orders | Vibhuti Patel | 5 | Open |
| Modify Login UI | Muhammed Khalid | 4 | Open |
| Create User Login Sessions | Muhammed Khalid | 5 | Open |
| Add mySQL ORM | Muhammed Khalid | 3 | Open |

An important part of this sprint cycle was to accomplish a complete product. In short, we wanted to have a complete product ready by the 25th, although we would still like to add much more. So, as a team, we worked hard to make sure that everything was working. We tackled the stories above and helped each other on the artifacts even though they were not directly assigned to us. Fortunately for this sprint, we completed all assigned stories.

These were the stories that we were able to finish in the second sprint. We did not have an issue in completing the artifacts as we learned from the first sprint of what we were capable of accomplishing within the deadline:

## Meeting 1 | Sprint Planning

#### Location: Lyndon Library 03/28

* Went back to our product backlog and moved to this sprint
* Also added stories from previous sprint
* Those stories have been rolled over. Hopefully, they can be accomplished in this sprint
* Broke up tasks and assigned to each respective team member best capable of performing them
* Also added estimates to each story (in hours).
* Final Sprint2 Backlog:

| **Stories** | **Assignee** | **Estimate (Hours)** | **Status** |
| --- | --- | --- | --- |
| Create UI allowing user to modify profile information | Vibhuti Patel | 2 | Open |
| Give Edit Permissions for Orders depending on User Type | Vibhuti Patel | 1 | Open |
| Add backend to User Submission UI | Muhammed Khalid | 5 | Open |
| Create UI showing list of orders | Vibhuti Patel | 5 | Open |
| Modify Login UI | Muhammed Khalid | 4 | Open |
| Create User Login Sessions | Muhammed Khalid | 5 | Open |

## Meeting 2 | Sprint Planning

#### Location: Online 03/31

* Still working on the same artifacts
* The Login UI was modified with CSS
* Learning about SQL ORM (SQLAlchemy
* Working on User Login Sessions

| **Stories** | **Assignee** | **Estimate (Hours)** | **Status** |
| --- | --- | --- | --- |
| Create UI allowing user to modify profile information | Vibhuti Patel | 2 | Open |
| Give Edit Permissions for Orders depending on User Type | Vibhuti Patel | 1 | Open |
| Add backend to User Submission UI | Muhammed Khalid | 5 | Open |
| Create UI showing list of orders | Vibhuti Patel | 5 | Open |
| Modify Login UI | Muhammed Khalid | 4 | Open |
| Create User Login Sessions | Muhammed Khalid | 5 | Open |

## Meeting 3 | Daily Sprint

#### Location: Online 04/04

* Still working on the same artifacts
* Everyone is making good progress on their stories
* Still working on the same stories from previous sprint meeting
* No new progress and no closed stories

| **Stories** | **Assignee** | **Estimate (Hours)** | **Status** |
| --- | --- | --- | --- |
| Create UI allowing user to modify profile information | Vibhuti Patel | 2 | Open |
| Give Edit Permissions for Orders depending on User Type | Vibhuti Patel | 1 | Open |
| Add backend to User Submission UI | Muhammed Khalid | 5 | Open |
| Create UI showing list of orders | Vibhuti Patel | 5 | Open |
| Modify Login UI | Muhammed Khalid | 4 | Open |
| Create User Login Sessions | Muhammed Khalid | 5 | Open |

## Meeting 4 | Daily Sprint

### **Location: Online 04/05**

* Progress the same and still working on artifacts
* No closed stories yet
* Vibhuti is still working profile data functionality and plans to work on today
* I am still working on Login UI modification and user login sessions and will continue till the next sprint

## Meeting 5 | Daily Sprint

### **Location: Cellphone 04/11**

* Good progress on artifacts
* Vibhuti worked on ajax calls to bridge JS and python code
* Vibhuti completed the profile information user modification functionality
* I implemented a SQL ORM using SQLAlchemy
* Created UI allowing user to modify profile information story is closed

| **Stories** | **Assignee** | **Estimate (Hours)** | **Status** |
| --- | --- | --- | --- |
| Create UI allowing user to modify profile information | Vibhuti Patel | 2 | Closed |
| Give Edit Permissions for Orders depending on User Type | Vibhuti Patel | 1 | Open |
| Add backend to User Submission UI | Muhammed Khalid | 5 | Open |
| Create UI showing list of orders | Vibhuti Patel | 5 | Closed |
| Modify Login UI | Muhammed Khalid | 4 | Open |
| Create User Login Sessions | Muhammed Khalid | 5 | Open |

* Create UI showing list of orders is closed

## Meeting 6 | Daily Sprint

### **Location: Olsen 404 04/20**

* A quick sprint meeting after class
* Muhammed is still working on the submission backend and user login sessions
* Vibhuti is working on creating edit permissions for orders
* No closed stories in this sprint meeting

## Meeting 7 | Sprint Meeting

### **Location: Online 04/24**

* Approaching the end of the sprint.
* Vibhuti has completed and closed are stories that were assigned to her
* Muhammed has also completed the stories that were assigned to him
* Opened a new story: Add mySQL ORM and closed in this sprint
* Currently working on a few bugs as we are doing a full integration with master
* Closed stories:

| **Stories** | **Assignee** | **Estimate (Hours)** | **Status** |
| --- | --- | --- | --- |
| Create UI allowing user to modify profile information | Vibhuti Patel | 2 | Closed |
| Give Edit Permissions for Orders depending on User Type | Vibhuti Patel | 1 | Closed |
| Add backend to User Submission UI | Muhammed Khalid | 5 | Closed |
| Create UI showing list of orders | Vibhuti Patel | 5 | Closed |
| Modify Login UI | Muhammed Khalid | 4 | Closed |
| Create User Login Sessions | Muhammed Khalid | 5 | Closed |
| Add mySQL ORM | Muhammed Khalid | 3 | Closed |

## Meeting 8 | Sprint End and Retrospective

### **Location: In class**

* Final demo will be done in class
* An overview and retrospective of the sprint will be discussed in class
* We have completed the project.

**Sprint Review and Retrospective**

We built upon what we had accomplished in the first sprint. In the first sprint we had a database schema created along with a very basic user login and registration. We also presented a demo of our prototype which demonstrated our progress. In the second sprint we accomplished much more. The original bland login UI was modified to look more pleasant and appealing to the eye using CSS and JavaScript. A list of orders was displayed to the application via ajax calls made in JavaScript. A backend support for user order submission and retrieval. And an implementation of a SQL ORM via SQLAlchemy for our MySQL database.

Regarding work that was not completed, we did not have an issue with this because all proposed work was accomplished within this sprint which is something that we were happy about. Also, no team member left the company. Unlike the last sprint, everything story was tackled in sprint2.

Because of this sprint, we have gained a confidence boost in what we are capable of doing. Our skillset in python, flask, and MySQL has vastly improved. We should try to tackle more stories because we were able to work efficiently in this sprint and we accomplished a lot with only 2 members in this team. Also, something that we mentioned in the first retrospective, is that we should make more time to meet in person rather than online. For this sprint, we met more online than in person. It did not impact our output, but it would have helped more with efficiency since we can talk out more issues in person rather than online. Due to limited time and projects from other courses, meeting in person was not very feasible.

For this sprint we could not think of anything that we should stop doing because we have had quite an efficient sprint this time around. However, one thing we should have done is split our sprint into two sprints because this sprint was about 30 days. We began the sprint back on the 28th of March, and currently it is the 24th of April. We should stop having longer sprints and keep it two weeks’ maximum which will allow for more efficient progress.

Things that we should continue doing is using Github and learning about other software practices. We should continue developing our product and continue coming with ideas for what we want to include in our product. How can we improve it and make it even more satisfactory to the user.