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:

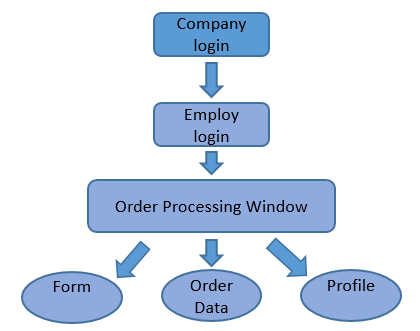
* Make it easy to calculate the expenses of each project or department for marketing and accounting team
* Save time by automatically notifying a project manager when a new order needs approval or notifying the department when an order has arrived
* Centralize ordering so the company doesn't need to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

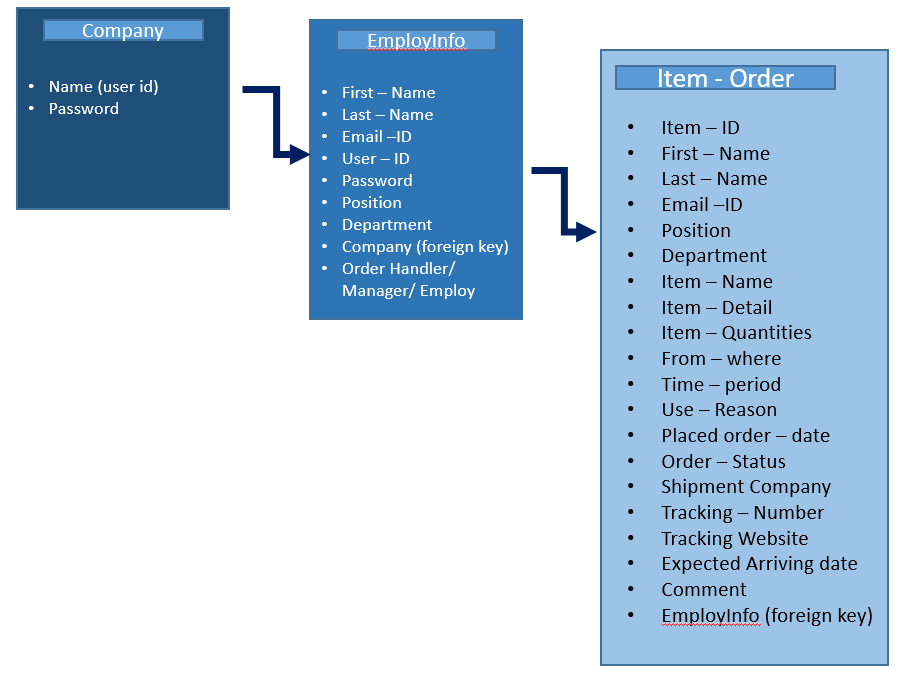
Our app will run in a web browser and consists of a user interface and a back-end database. The application works in two parts: (1) Requesting new order by completing a digital order form (2) Table which contains a list of all the orders that exist within the company. In addition to checking and keep 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.