TABLE OF CONTENTS

---

1. Executive Summary

1.1 Project Overview

Coffee shop chains are become more ubiquitous in Albania, with at least three of them being fairly prominent and well-heard to the average coffee consumer. In light of this, we can consider the possibility that a management system for this group of businesses may be in fact in demand.

Creating such a management system would require a division between the individual administration of single stores, as well as the 'back end' of the system, regarding management of the whole chain. Another divide is between the service portion (which will involve the above two sides) as well as the supply and storage portion, dealing with the storing and delivery of materials and supplies needed for each store.

Coordinating each individual portion of the management system would be the greatest challenge of the management system, not to mention the need to research legislation, economics and general business management, in order to properly create the system. Beyond that, a number of schematics and diagrams will be needed to properly create an idea of the system.

1.2 Purpose and Scope of this Specification

In Scope of these requirements:

-Taking into consideration various uses of the project by three categories of system users (Workers and Managers, Financial workers and General Administrators).

-Dealing with supply and distribution of needed materials for the business that is served by the application, to conduct itself properly.

-Keeping with legislation and other articles of law which mainly concern:

-Automatic calculation of tax payments

-Data protection of individual users, which involve employees of the company.

-Managing the flow of work and presenting a clear canvas of information and statistics regarding the service, in assistance of making better informed executive decisions

Out of the scope of these requirements:

-Handling the management of the supply and storage, as well as delivery. This would require it's own, separate magazine management system.

-Handling equipment and their distribution: Purchase of computers, electronic registers as well as machinery that aids daily work (e.g coffee machines) is an administrative decision, that the system cannot be made responsible for.

-Handling the hiring of new employees, as well as their firing. The system cannot and will not be able to automatically hire or fire employees.

2. Product/Service Description

2.1 Product Context

Being a web-application, the product is well-capable of interfacing with the web as a whole. Therefore, the possibility for integration into a system that is beyond the scope of the product itself, is in fact present.

With the above consideration, it must still be noted that the product is intended for somewhat isolated use, and no unconcerned users should be allowed to take part in it. It is dedicated to specific employees of the coffee shop chain service that employs it (whether it be one or many, and with modifications per case).

2.2 User Characteristics

User profiles, are listed as below. There is a total of four, although for administrative purposes, the first two will be under the same group of functional requirements (Note: This does NOT mean they will operate in the same way)

User Profile: Shop Worker

Typically not too involved in the larger organizational processes, but will probably be the type to peruse the product the most. Little expertise requires, barring the use of a simple online interface, as well as general adequacy as a shop worker and waiter. Little technical expertise needed, however familiarity with the functionalities of the system is needed, so as not to cause problems.

-The Shop Worker has access to the shop worker interface:

-The Shop Worker can log in, log out, and change their own password.

-Create orders, and mark their completion

-Mark the amount of money acquired, via payment for orders completed.

-Note any amount of supplies used in the process, and potentially ask for emergency supplies.

User Profile: Shop Manager

Greater responsibility in the bigger picture of the system, as they need to manage payment for individual workers, as well as periodically check for supplies. Supply requests may be made by the system, or the shop workers, however reports are filed by the manager, and they also possess the ability to do so.

Thus, while much like the shop worker, little technical expertise is needed, bar that required to use a simple web-application, understanding of the workings of the company and system as a whole, are a requirement.

-The Shop Manager has their own interface:

-Logging in, logging out, registering and deleting local employees, change their own password.

-Checking on present store supplies

-Checking on status of workers (but NOT on personal worker data)

-Checking employee reports, that are automatically generated, monthly.

User Profile: Economist

Handles most of the financial situation of the system. Familiarity with the system as a whole is a must. Familiarity with the workings of the company, as well as handling every portion of the system, not to mention being able to foresee potential problems coming from other sectors and the need for some technical expertise regarding potential problematic cases with their own (financial) section of the product, are much preferred.

-The Economist has an interface regarding the whole system:

-They can log in, log out and change their own password.

-They can check the economic status of every shop in their spread of influence.

-They can generate reports, and approve or disapprove budgets and supplies for individual shops.

User Profile: System Administrator

The requirements for this type of user are the steepest.Their duty is to manage the system as a whole. Technical knowledge of managing the system and potential errors is requires, seeing that the system

administrator can often be the company's IT as well. They should be familiar with the structure of the system database, as well as the functionalities provided by the system.

Beyond that, some financial knowledge should be needed, to be able to properly interact with the economical section of the product. Awareness of most user duties and functions, as well as access to the system.

-The System Administrator has a fairly large access to the system:

-His interface allows for logging in, and registering new users of any level in the organisation.

-It allows the administrator to access the balance and status of every individual shop and the general status of the system

-It allows them to read monthly reports from individual branches and the economist

-It allows them to remove or add new users, as well as add new products

-Check the supply status, and communicate with the storage (might not be implemented in the system, depending on business details)

2.3 Assumptions

It is assumed that the product users have a persistent network connection, otherwise the product will be unavailable. Following this, it is also assumed that the company can provide electronic registers, computers and other platforms that are requires of the system. Following this, it is assumed still that the browsers are sufficiently updated. While the product will try to cover for all possibilities, a more modern browser is preferred for work to be smoother.

2.4 Constraints

The product might require:

-Log file keeping. This is for purposes of an electrical outage, or other system errors.

-Security. Being a web application, difficulties with keeping the program secure may arise.

-There are few limits on disk space or memory, however an internet connection is required.

-Maintaining the database will also be required of the database administrator of the system, most often the IT specialist.

-Frameworks and programming languages associated with the product, will have to be web-based.

2.5 Dependencies

-A front-end/interface needs to be constructed first and foremost, possibly in parallel with the database.

-Following that, a diagrammatic view of the system needs to be constructed, in line with the database format. The back end of the product will thus be defined.

-The system needs to be online, and generate automatic reports, regarding financial status, which are calculated via the related algorithms.

-Individual shop modules need to be defined properly, and distributed to each shop. They must all connect to a central module.

-These can be built in parallel, however the individual shop modules are more fit for building first.

-A form of connection must be established between the administrative section and the daily work section of the product.

-A form of separation must also be but in place, so that each module is independent.

-Security must be added to the collective modules of the product.

3.1 Functional Requirements

CS_GR(Coffee Shop_General Requirements) - Requirements relating to
CS_IS (Coffee Shop_Individual Store) - Requirements relating to individual coffee stores
CS_F (Coffee Shop_Finance) - Requirements pertaining to finance
CS_SA (Coffee Shop_System Administration) - Requirements pertaining to the administration.

Req#	Requirements	Comments	Priority	Date Rvvd	SME Review/Approved
CS_GR #1	There should be separate views for separate user levels	*Specifically, four have been outlines	1	3/29/2019	
CS_GR #2	Every user should be able to log in and out		1	3/31/2019	
CS_GR #3	Every user should be able to access their own part of the system		1	3/31/2019	
CS_IS #1	The system should have profiles for each worker		1	3/29/2019	
CS_IS #2	The system should be able to generate reports for each worker		2	3/29/2019	

CS_IS #3	Worker profiles should be private.	*Data protection	1	3/29/2019	
CS_IS #4	Managers should be able to read all worker reports		2	3/29/2019	
CS_IS #5	Managers should be able to register new employees		2	3/29/2019	
CS_IS#6	Managers should be able to file a supply balance report, and request for supplies at the end of the day.		2	3/29/2019	
CS_IS #7	Workers should be able to make an emergency request		2	3/29/2019	
CS_IS #8	A statistical monthly report, generated weekly	It would help improve the product supply, adjusting it as per preferences.	3	3/29/2019	
CS_SA #1	The admin should be able to view all other users		1	31/3/2019	
CS_SA #2	The admin should be prohibited from viewing user private information		1	31/3/2019	
CS_SA #3	The admin should have access to all product tables		1	31/3/2019	
CS_	The admin should be		2	31/3/2019	

SA #4	able to register or remove users, of any lower level				
CS_ SA #5	The admin should be able to edit, add or delete new products		2	31/3/2019	
CS_ SA #6	The admin should be able to view all reports, from all users		1	31/3/2019	
CS_ SA #7	The admin should be able to view the economic report/balance delivered at the end of the month		1	31/3/2019	
CS_ SA #8	The admin should be able to view a statistical overview of the system		3	31/3/2019	
CS_ F #1	There should be a daily report for each shop	*Variable report time, depending on implementation or individual store.	1	1/4/2019	
CS_ F #2	There should be an automatic tax calculation procedure, monthly.		1	1/4/2019	
CS_ F #3	An economical report should be auto-generated	Note that this is different from audit tracing	2	1/4/2019	

	monthly				
CS_F #4	There should be a file that tracks every finished transaction. Files are preserved up to a certain period.		1	1/4/2019	
CS_F #5	There should be a calculation of business expenditure, concerning supplies		1	1/4/2019	
CS_F #6	The system should have a calculation of total monthly earnings, done automatically		1	1/4/2019	

3.2 Non-functional Requirements

3.2.1 Product Requirements

3.2.1.1 Placeholder

There will be four separate interfaces.

- Shop Employee Interface:
    - It should primarily contain a log-in window, with the option to change password. The exact password implementation details can change over time.
    - It should then contain an user profile, with the option to view the following:

        - Orders: A list of orders pending, and one of orders completed
        - Report: A daily or weekly (adjustable) report, which is
        - Log Out: The option to log out of the site.
        - Emergency Supply Request: Send a notification to the main HQ to send supplies of a certain product that is needed.
- Shop Manager Interface:
    - It should allow for logging in and registering new workers, locally
    - It should be able to allow one to authorize password changes, if available.
    - It should allow the manager to check:
        -His own profile
        -User reports, either daily or monthly
        -Product supply status, and a balance for the day (financially speaking)
        -Log out of the site.
        -Request supplies, as per the entry in Shop Worker interface.
- Economist Interface:
    - It should allow for logging in.
    - It should allow for viewing of monthly balance report, of the local shops.
    - It should however, restrict this to the ones they have on their area.
    - Able to compile (automatically, or with adjustments) a general report.
    - Able to make requests and send notifications to individual shops or administrator.
- Administrator Interface:
    - Can keep track and observe all reports.
    - Can add new users, and promote users to managers. Can also remove users from the system.

- Can manage requests and notifications from individual shops and economist.
- Can log in, and log out.

3.2.1.1 Usability Requirements

-There should be an extensive document that provides instruction on all of the potential uses of the system.

-This document should be used to instruct users on all methods of accessing the system.

-The product should be fairly easy to utilize.

-The documentation should be split in modules, concerning each type of user of the system.

-The application should be secure, to users, and they should be confined to their areas of interest only.

-The application should possess clearly visible functionalities, and allow for easy access.

3.2.1.2 Efficiency Requirements

3.2.1.2.1 Performance

Being a web application, a lot of the performance depends on the assumption about Internet connection, specifically pertaining to what the ISP offers in terms of speed and quality of connection. This is a primary concern regarding the product.

3.2.1.2.2 Space

The maximum number of users is variable, and it is doubtful that the company will have more than the system can support. It is predicted, that at most, several hundred (no more than 500 employees) will have to interface with the application, as several functions of the company will be separate from the system.

3.2.2 Dependability Requirements

-The product should be available at all times, online.
-The product should be available only to the concerned parties.
-If external parties are concerned, an informational page can be created, to interface with them.
-There should be little issue with geographical spread, seeing that it is a web-app.
-Downtime impact varies, depending on when the downtime occurs. Downtime outside of working shifts is acceptable, and can be repaired. Downtime at a critical time that serving is required, can be more problematic.
-Scheduled maintenance can be left for times when the store itself is closed, or alternatively times with as little traffic as possible. Off note is the fact that scheduled maintenance should be restricted to a certain time-frame as not to cause too many issues.
-There should be as little downtime as possible during working hours. At most one occasion per hour is acceptable, even if highly undesirable.

3.2.1.3 Security

Considering the structure of the system database, there will be difficulties arising with the product menu and the expenditure of individual components of the products, regarding automatic calculation of resource consumption, and supply requesting, as well as potentially statistics. It is unlikely that the total combination of ingredients per drink, taken individually (per each product they are part of) will exceed 20,000, which is a tested value regarding system performance (regarding queries).

3.2.1.3.1 Protection

The system will undergo several layers of protection:

-Activity logging, as well as error logging
-Security encryption, for passwords.
-User input filtering, in any case an user enters undesirable data
-SQL filtering, in case of SQL Injections.
-Usage of prepared statements, for the same reason as above.
-Usage of proper methods of cross-site data transfer.
-Session refreshing, for Session ID hijacking attempts.

3.2.1.3.2 Authorization and authentication

-Only users who are members of the company/employed under the company should be able to view the system.
-Users of a certain tier should only be able to view the concerned architecture.
-Every user logs in with a company provided username, and a random password, which they can change using a company provided private code.

3.2.3.3 Latency

-The interface linking and the placement within the collective of the application of individual pages, should be as optimal as possible, with as few jumps as possible between two pages, to avoid latency.
-Queries should be optimized, as should be algorithms that generate values (often for calculation of taxes and even for report generation) and documents for the system.
-Connection based latency is not covered by the system.

3.2.2 Organizational Requirements

3.2.2.1 Environmental Requirements

The users should first and foremost have internet access. They should be able to access the app, securely.

Users should be able to operate properly, both by accessing the system itself, to view the menu and select proper products, viewing their own orders, or using the database (not directly!) to view the whole list of products.

The users implicitly affect the database, via completing orders, which reduce a standard amount out of the total available at the beginning of the day.

The users have their reports automatically sent to higher-tier users.

3.2.2.2 Operational Requirements

The only users that are allowed by managers or the system administrator are capable of using the product.

Users must first authenticate themselves via logging-in. Following that, they can change the password only via a personal, randomly generated code, given at the beginning..

Users should be able to notify different tiers of the administrator, the economist receives monthly reports generated automatically. Shop managers receive

3.2.2.3 Development Requirements

Creation of the database is what comes first in the development process.

Communication with the customer, should be handled at every stage. Once a step of the development has been completed, the communication should be restored.

The development follows the spiral model in this regard. Front end and back end as well as individual interfaces are handled as time goes on. Following that, until the product is improved, the spiral model cycles through.

Each interface has a different customer base to communicate with, normal workers, management positions, economists and the administrator of the system. The later holds onto the whole of the system, and is more involved with it’s creation, as well as far more specialized, regarding the functionalities associated with the product.

Testing phase is where the product is presented to a group of users, and communication is finalized, final touches are made and the product is launched. The product can be updated however, and launched once again as long as the structure of the database is not altered.

This stems from the consideration that while new technologies and optimizations are bound to occur, changes in the business model are highly unlikely, and therefore the structure of the company reflected in the database system is too, unlikely to undergo alterations.

3.2.3 External Requirements

3.2.3.1 Regulatory Requirements

System Interface/Integration

The system should be capable of running on most Web-Browsers. While technologies accomodate for most modern browsers, such as Mozilla or Chrome, some owners may opt for Internet Explorer, in which case alterations may need to be made.

Alternatively if the user can be communicated, and the decision to use only the advanced browsers is agreed upon, work can continue far more simply.

Typically, specifications depending on the computers (and therefore Operating Systems) available to the company, which are not the responsibility of this product development team, but of the company itself, would help tremendously.

Communication is continuous and the conclusions drawn by it are replenished with each step of development.

3.2.3.1.1 Network and Hardware Interfaces

Persistent TCP connection required, for connecting the various sites of the app.

Operating System is of little relevance, as long as it supports browsers such as Google Chrome or Mozilla.

Typically, company-approved devices are the only ones with proper access to the system. Network card is a necessity for the application to function.

A server should be up at all times, to host the system, as should a connection to the database. These need almost full uptime, and should be down only in the event of maintenance.

Access and error logs should be created and maintained, for everyone who accesses the site, these will exist within the server.

Connection to an electronic register optional, and the implementation exceeds the scope of the project. Orders will be preserved by the application.

Receipt printer will however, need to be implemented. This is in accordance with the laws of the state of Albania, detailing the receiving of a tax coupon upon the completion of a monetary transaction.

3.2.3.1.2 Systems Interface

The Worker Interface will send a weekly report to the Manager Interface. The report will contain the following data, arrangement at the discretion of the programmer and of the party requesting the product, following basic tenets for clarity and readability:

-Worker id, name and surname.

-Date of report compilation.

-Number of orders completed.

-Number of days present at work, with working hours for each day, collected upon log-in and log-out.

-Sum of money produced via the above orders. Automatically calculated.

The Manager Interface produces a monthly report, which is sent to the Economist Interface. It contains the following, abiding by the above point:

-Shop id, Manager id, name, surname of manager, as well as address of shop.

-Date complied, month it belongs to.

-Total earnings per month.

-Total resource expenditure.

-Total tax associated with the monthly balance.

The administrator has the right to read all of the above reports. He receives notifications whenever a report is produced, and has them separated into managerial and financial reports. They are the same as the above.

The manager can also view a total list of products, available at his store. The administrator has access to all products available at every shop as well.

Assignment of pay is done from the managerial interface, for workers, and they can view it from their own interface.

Assignment of pay for economists, and managers is handled by the administrator.

3.2.3.2 Ethical Requirements

The product cannot make decisions or act on it’s own regarding any of the following:

-Adding or removing a worker from the system.
-Making changes to payroll database.
-Making supply calls on it’s own, without user approval.

Children’s Privacy

The software and its services are not directed to anyone under the age of 13 or 18 (work age). The Site does not knowingly collect or solicit information from anyone under the age of 13 or in some cases under 18, or allow anyone under the age of 13 to use the software servicers. And in any case if that information of persons under necessary age is collected, then

this information will be automatically deleted from server, database and so on where we will delete that information as soon as possible.

3.2.3.3 Legislative Requirements

3.2.3.3.1 Data Protection

The state of Albania has passed law no.9887, “On the protection of personal data” in 2018, which allows for an independent institution to oversee the protection of personal data. This affects the users of the product as well, not to mention it’s structure: no user of whichever tier may be able to view any sensitive information, or seek to acquire it by any means.

This is made sure of by the system. It should be noted, that names, surnames and other information that is provided by the employer, upon hiring does not count for these purposes, as it is common sense for it to be known. Passwords and private profiles, should however be as the term suggests, private.

3.2.3.3.2 Taxation Laws.

As of 2014, Albania makes use of a three-bracket progressive income tax. It is likely that the subjects interested in the product, will be of the third bracket, and thus be taxed 23% of their earnings.

Following that, the VAT (Value Added Tax) or as it is known in Albania, TVSH, will also have to be taken into account. It’s value is 20%.

Social Security tax, which includes health insurance contributions has to be taken into account as well. Social security and health insurance contributions are paid on employment, civil and management income. Contributions is paid on a monthly income, from a minimum of 22,000 to a

maximum amount of 95,130. The employees contribution is 9.5% while the employer pays 15%. Health insurance is levied at 1.7% for both employee and employer. The self-employed pay 23% for social security and 7% for health insurance.

Transactions concerning the payroll should be properly audited as well, a trace log should include a description of all the transactions, as well as a timestamp, to make sure nothing is lost in the event of system errors.

3.2.4 Other Requirements

3.2.4.1 Data Management

The database will contain a number of tables, and the following information, which will also be depicted in an ER diagram:

3.2.4.2 Portability

The system should be highly portable by virtue of it being a web-application. Barring the server, distribution of the system on a per-shop basis, containing the Shop Worker and Shop Manager modules, which have been outlined previously.

The main headquarters of the business concerned, should be where the Economist and Administrator modules are seen, and distributed in, as well as where data is stored.

It is prudent to communicate with the company and observe the specifics of their system, should they possess more modern browsers, portability is

less of an issue, as special coding arrangements have to be made for older browsers.

3.3 Domain Requirements

-The database should accommodate for the following concepts related to the nature of the product:

-User profiles, where users are defined as employees
-Reports of two different fashions on work and economic progress.
-A menu of products, which can be modified for fuller automation.
-Continuing off the above, a link between products and their constituents.

-The tax payment calculation algorithm should be adjusted to the specifics of the business, as outlined in section 3.2.9, part 2.

-The online nature of the application should be fitted with security measures, as it is one of the less safe choices concerning attacks.

-Further still, the online nature of the application will require appropriate means of connecting various modules of the whole system.

4. Use Cases/Scenarios

Business Scenario #1:

Order placing.

Actors: Shop Worker, Customer (not part of the system).

Start state:

Shop worker is logged into the system and has received an order (not part of the system yet).

Scenario flow:
-An order is created out of one or more menu items.
-Menu items are then taken apart into their constituents, of standard quantity.
-Order is then added to the interface (but not the database) automatically by the system. (Work may not begin in completing it, however this is outside the system scope)

End State:

Order has been added to the interface

Business Scenario #2

Order completion

Start state: User is logged in, at least one order exists in the interface.

Scenario flow:

-Order is processed, and is deemed complete when delivered to the customer.
-The individual quantity of products is subtracted from the amount present at the shop.
-The order cost is added to shop earnings and written in the day log.
-The order is removed from the interface, after all concerned attributes have been updated.

End State: Particular order is deleted from system

Business Scenario #3:

Report Creation.

Actors: System, Shop Manager

Start state: The due time set by the employer,for the report to be generated is reached.

Scenario Flow:
-The daily report for all order proceedings (in total of products used and income) is produced.
-It is sent to the manager to review it.
-It is further sent to the administrator, as an aside, to review at one's leisure if needed.

End state: Report is created and printed, and it can further be transmitted to other users.

Business Scenario #4:

User Registration.

Actors: User of any level barring Admin, Admin, Manager.

Start State: An administrator is logged in the system, either of the HQ or individual shop.

Scenario flow:
-A new user enters the system.

-If it is a shop worker, they are registered by the manager under their name, surname and an username conforming to the company standards.
-A shop manager or economist is added to the system by the system administrator, through the same procedure
-User is given a randomly generated or standard password, and a personal code

End state: A new user is added, with all the required information, or addition is denied.

Business Scenario #5

Password Change

Actors: Any user of any level

Start state: User is logged in, using his original password.

Scenario Flow:
-User accesses the password change interface, with their other login details (username)
-User inserts their fixed, immutable personal code.
-User changes password (reconfirming it a second time), upon confirmation of personal code
-Upon failure of confirmation, user is re-prompted
-Upon failing several re-prompts, user is locked out of password-resetting for increasingly longer intervals of times.

End state: Password is successfully or unsuccessfully changed. In either case, the interface is left.

Business Scenario #6

Log In

Actors: Any user of any level

Start State: User enters the system

Scenario flow:
-User accesses the system, finding the log-in interface.
-User inserts their credentials, username and password.
-Upon confirmation user is allowed to access their concerned part of the system.
-Upon failure of confirmation user is re-prompted
-Upon failing several re-prompts, user is locked out of the system for increasingly longer periods of time.

End State: User is logged in, or alternatively forced out of the system due to failing the log-in check.

Business Scenario #7

Requesting Supplies

Actors: Shop Manager, System Administrator

Start State: Individual shop manager or administrator is logged in, and accessing the unique supply request option.

Scenario Flow:
-Upon the manager's discretion, which can be at any given time or alternatively, upon receiving a warning from the system, a request is sent to the company headquarters
-The Administrator receives the request, and authorizes it.

-Supplies are sent to the store.
-The notification is deleted

End state: After both sides acknowledge the request and transaction, action is taken.

Business Scenario #8

Creating Monthly Report (Economic)

Actors: System, Economist

Start State: Economist is logged in and on the report interface

Scenario flow:
-At the end of each month, the system generates a report
-It contains total earnings, total spendings per shop, and is sent to the economist
-It is then added with total taxes to be paid, calculated automatically
-It is further supplied with wages, to each worker
-The total is written at the end of the report

End state: The report is printed out and appears on the screen

Business Scenario #10

Logging Out

Actors: Any user

Start State: User has to be logged in

Scenario flow:

-The user selects the Log Out option
-The user then logs out

End state: User is logged out

Business Scenario #11

Removing an user from the system

Actors: Administrator, User of any level

Start State: Administrator is logged in and viewing the user list.

Scenario flow:
-An user leaves company employment
-The user is removed from the system by the admin
-The admin does this by accessing the interface allowing for user removal, and deleting the user from the database

End State: User is confirmed to be deleted from the system.

Business Scenario #12

Adding a product to the menu

Actors: Administrator

Start state: Administrator is logged in and on the proper interface.

Scenario flow:
-The administrator opens up the item list
-The administrator inserts all the attributes required to form a new item in the appropriate fields.

-The administrator confirms the addition of the new item
-Attributes are extracted from the fields, and item is added to the database

End State: Product addition confirmed.

Business Scenario #13

Adding a new product category

Actors: Administrator

Start state: Administrator is present at the proper interface

Scenario flow:
-Administrator creates new category via proper window
-Administrator then adds the category id to the products that fit in it, or updates it properly
-Administrator can potentially automate this process if attributes of the products belonging to the new category have something in common, relating them to said category

End state: Product category addition confirmed.

Business Scenario #14

Deleting a product

Actors: Administrator

Start state: Administrator is logged in and is using the interface that allows him to delete the product

Scenario flow:

-Should the need arise, the administrator can delete a product that is no longer being offered
-Simply entering the id should be enough to delete the product from the database

End state: Product is confirmed to be deleted

Business Scenario #15

Deleting a category

Actors: Administrator

Start state: Administrator is logged in:

Scenario flow:

-Should the need arise, the administrator can remove a category
-Upon removal (via id insertion), the items belonging to the category have their category id set to 0
-Care should be taken to properly re-distribute them to a new category.

End state: Deletion is confirmed.

Business Scenario #16

Security Violation

Actors: Any low-level user or admin

Start state: Security violation detected.

Scenario Flow:

-If detected by low-level user, user contacts admin, admin then tries to resolve the security measure by himself and current security applications in disposal. If he cannot do so by himself and the system is still compromised, admin contacts cyber-security professionals to resolve the problem.

-if detected by admin he tries to resolve the security measure by himself and current security applications in disposal. If he cannot do so and the system is still compromised, admin contacts cyber-security professionals to resolve the problem.

-Admin may also choose to report the violation to respective police authorities.

End State: Security violation is dealt with and system is safe and running, no data is compromised.

Business Scenario #17

Privacy Violation

Actors: Any low-level user or admin

Start state: Privacy Violation detected

Scenario Flow:

-If detected by low-level user, user contacts the admin, admin in response tries to determine the level of privacy breach and tries any response measure he has in disposal in the current event. He then contacts the cyber-security professionals in order to restore any lost data from servers and prevent the privacy attack from doing any more damage to the system.

- if detected by admin , he tries to determine the level of privacy breach and tries any response measure he has in disposal in the current event. He then contacts the cyber-security professionals in order to restore any lost data from servers and prevent the privacy attack from doing any more damage to the system.

-Admin may also choose to report the violation to respective police authorities.

End State: Privacy violation is dealt with and system is safe and running , any data loss is recovered.

Business Scenario #18

Recovery/Maintenance break

Actors: Admin

Start state: System is under maintenance or under recovery from attack.

Scenario flow:

-The system will be regularly tested and updated by professionals which states in itself that the system will be offline during the period. Maintenance breaks will occur during pre-discussed time intervals between admin and system testers or developers. No one can access the system during this time period which is known amongst all users.

-If system has been under attack and as a result was offline, it needs recovery time in order to come back online and running. During this time period no one can access the system and should only do so when system comes back online.

End State: System is back online and users may continue their work as normal.

APPENDIX

Appendix A. Definitions, Acronyms and Abbreviations.

Management System: The product itself is defined as a management system, it facilitates the overall management of the business concerned to purchase it.

Schematics/diagrams: Visual representations of the workings of the system, displaying parts of it’s function.
Web-application: A product or application, of the client-server model, which the client runs on his own browser, and the server is run remotely.

Log-in: The ability to access the product, via a given identification. Program functionality is locked behind logging in.

Log-out: Leaving the product, via the opposite action of letting out.

Register: Make someone a part of the system, by adding their records into the system database.

Database: A collection of structured data, related to the system, involving a list of workers, products etc.

Module: Each of a set of standardized parts or independent units that can be used to construct a more complex structure, such as an item of furniture or a building.

Network: Two or more connected computers.

Appendix B

-The Privacy Policy document, regarding the product, compiled by Baftjar Tabaku, contains the legal and regulatory requirements as well as clarification on other cases.

-Law No. 9887 for data protection, approved in 2018.

-Albanian taxation laws and regulations, regarding:

- -Income tax
- -VAT
- -Social security tax

-Wikipedia and dictionaries, concerning Appendix A, for definition of terms.

-Documentation of Web programming languages, such as php and JavaScript, concerning adaptability to various browsers (prime example being Internet Explorer)

-Various projects of the earlier years, concerning structure

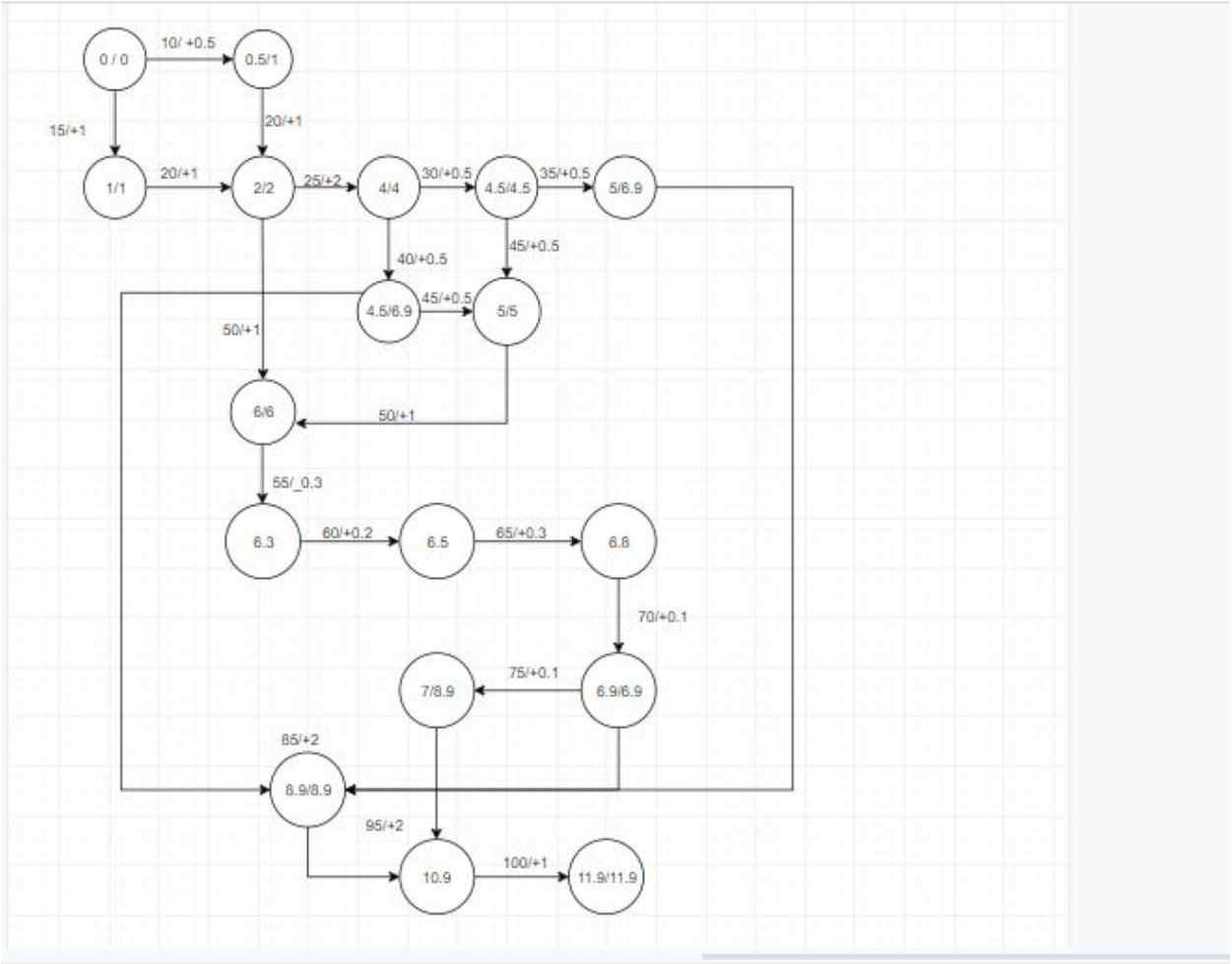
-Document that explains the Requirements Specification, which this work follows after.

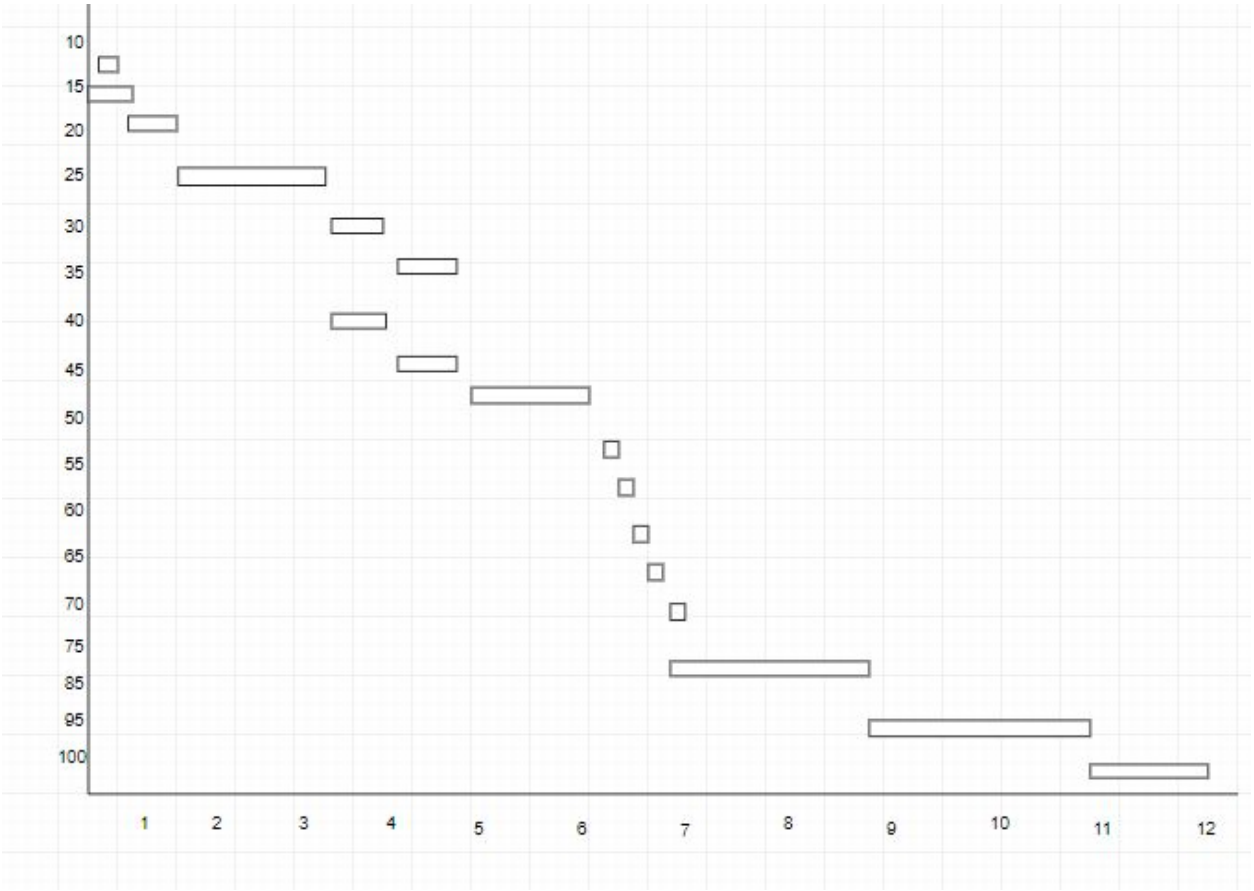
Appendix C

Network Analysis

	Activity	Duration	Depends On
10	Project Analysis	0.5 week	-
15	Project Research	1 week	-
20	Requirements analysis	1 week	10,15
25	Requirements engineering	2 weeks	20
30	Use Case Diagram	0.5 weeks	25
35	Activity Diagram	0.5 weeks	30
40	State Diagram	0.5 weeks	25
45	Data Flow Diagram	0.5 weeks	40,25,30
50	ER Diagram	1 weeks	20,45
55	Object Diagram	0.3 weeks	50

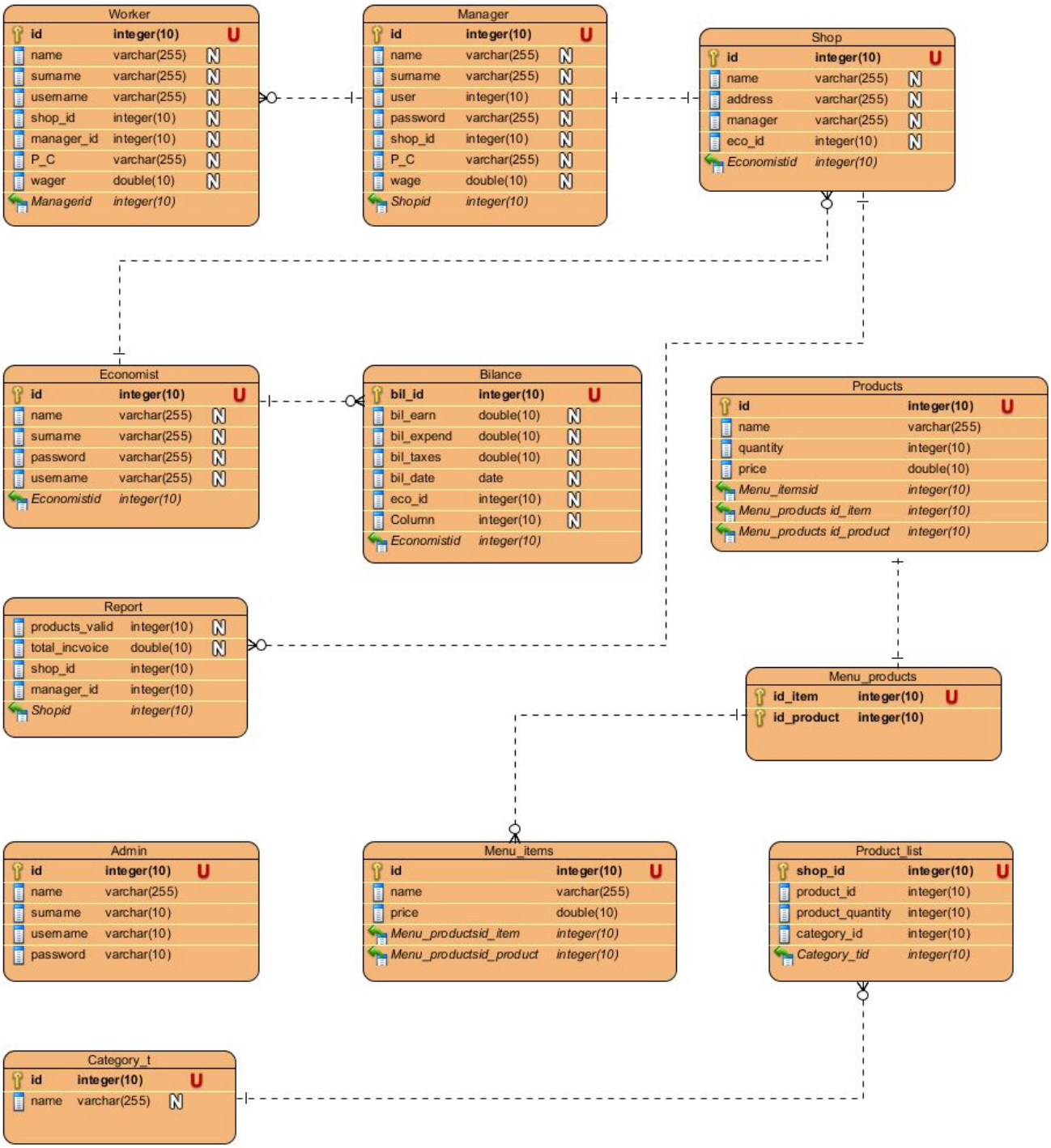
60	Class Diagram	0.2 weeks	55
65	Collaboration Diagram	0.3	60
70	Component Diagram	0.1 weeks	65
75	Deployment Diagram	0.1 weeks	70
85	Front End Coding	2 weeks	35,40,70
95	Back End Coding	2 weeks	75,85
100	Testing	1 week	85,95


0 / 0
10/ +0.5
0.5/1
15/+1
20/+1
1/1
20/+1
2/2
25/+2
4/4
30/+0.5
4.5/4.5
35/+0.5
5/6.9
40/+0.5
45/+0.5
4.5/6.9
45/+0.5
5/5
50/+1
6/6
50/+1
55/_0.3
6.3
60/+0.2
6.5
65/+0.3
6.8
70/+0.1
7/8.9
75/+0.1
6.9/6.9
85/+2
8.9/8.9
95/+2
10.9
100/+1
11.9/11.9

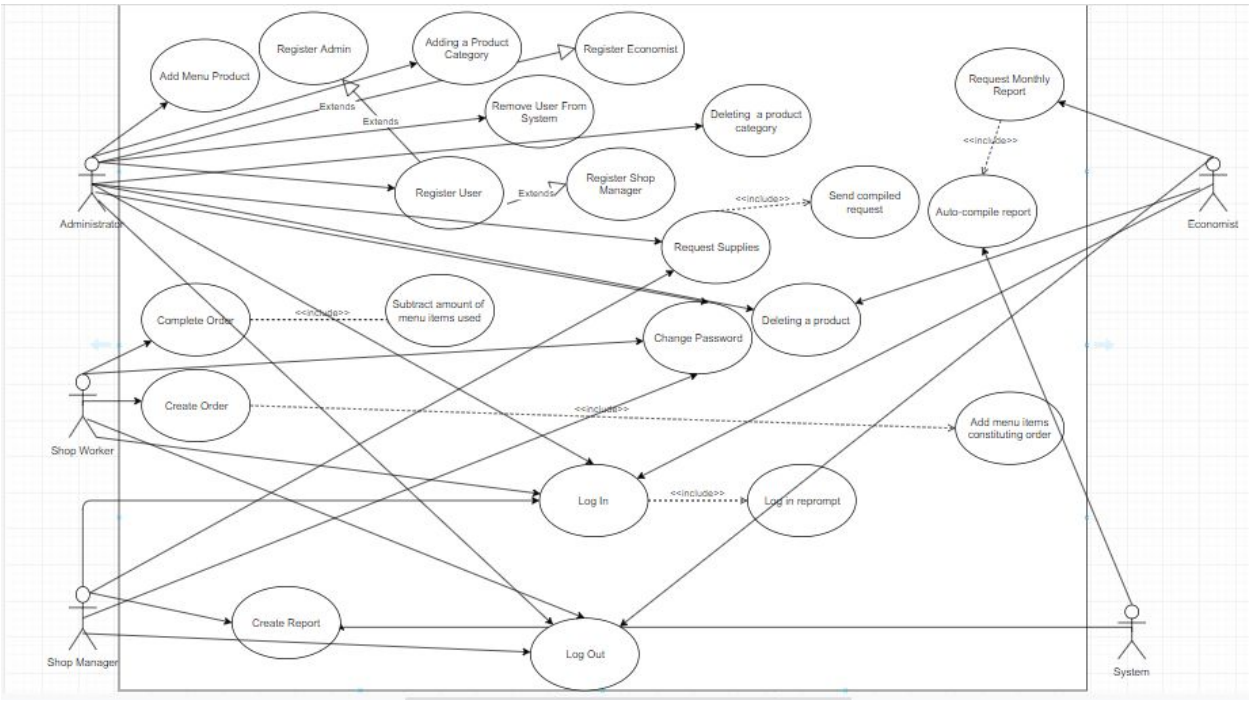

10
15
20
25
30
35
40
45
50
55
60
65
70
75
85
95
100
1
2
3
4
5
6
7
8
9
10
11
12

Appendix D

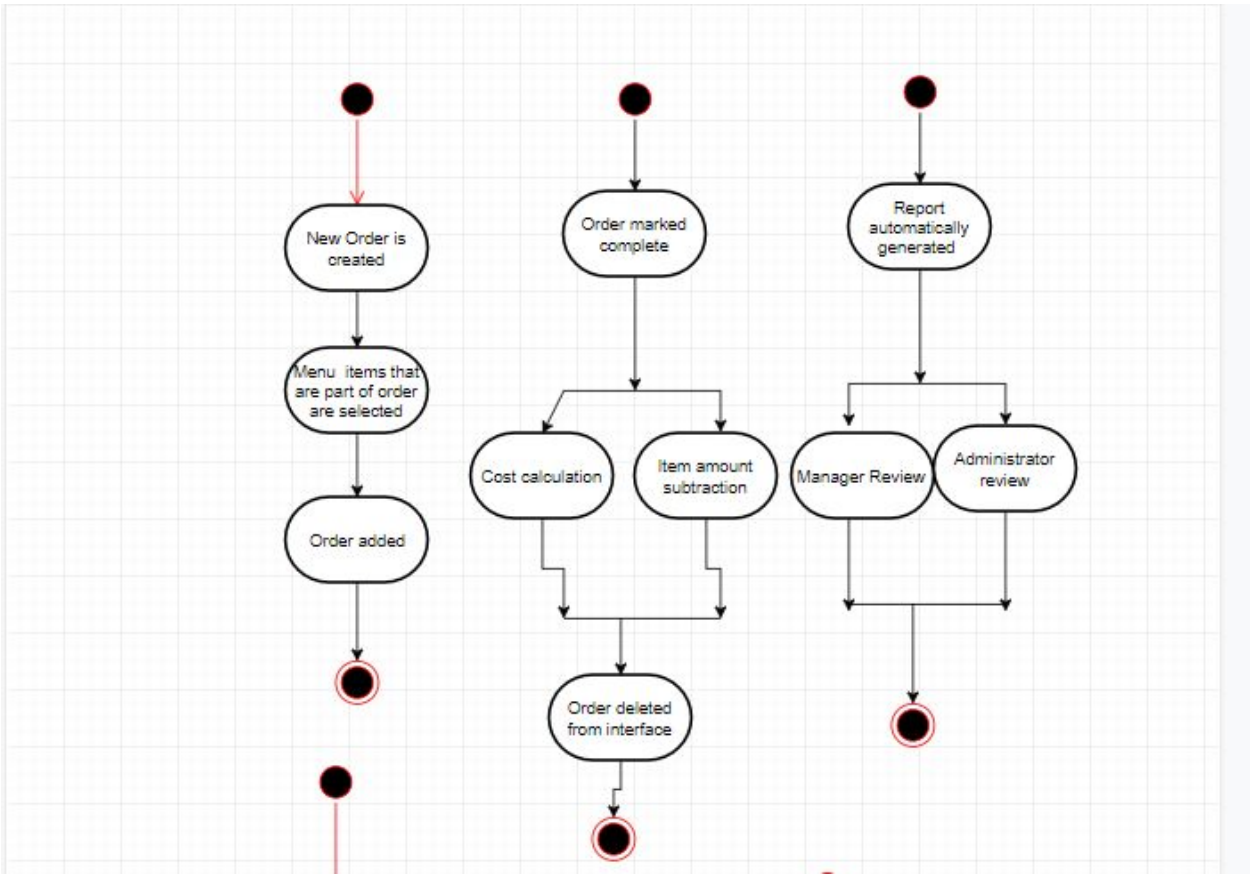
Cappuccino Database ER model



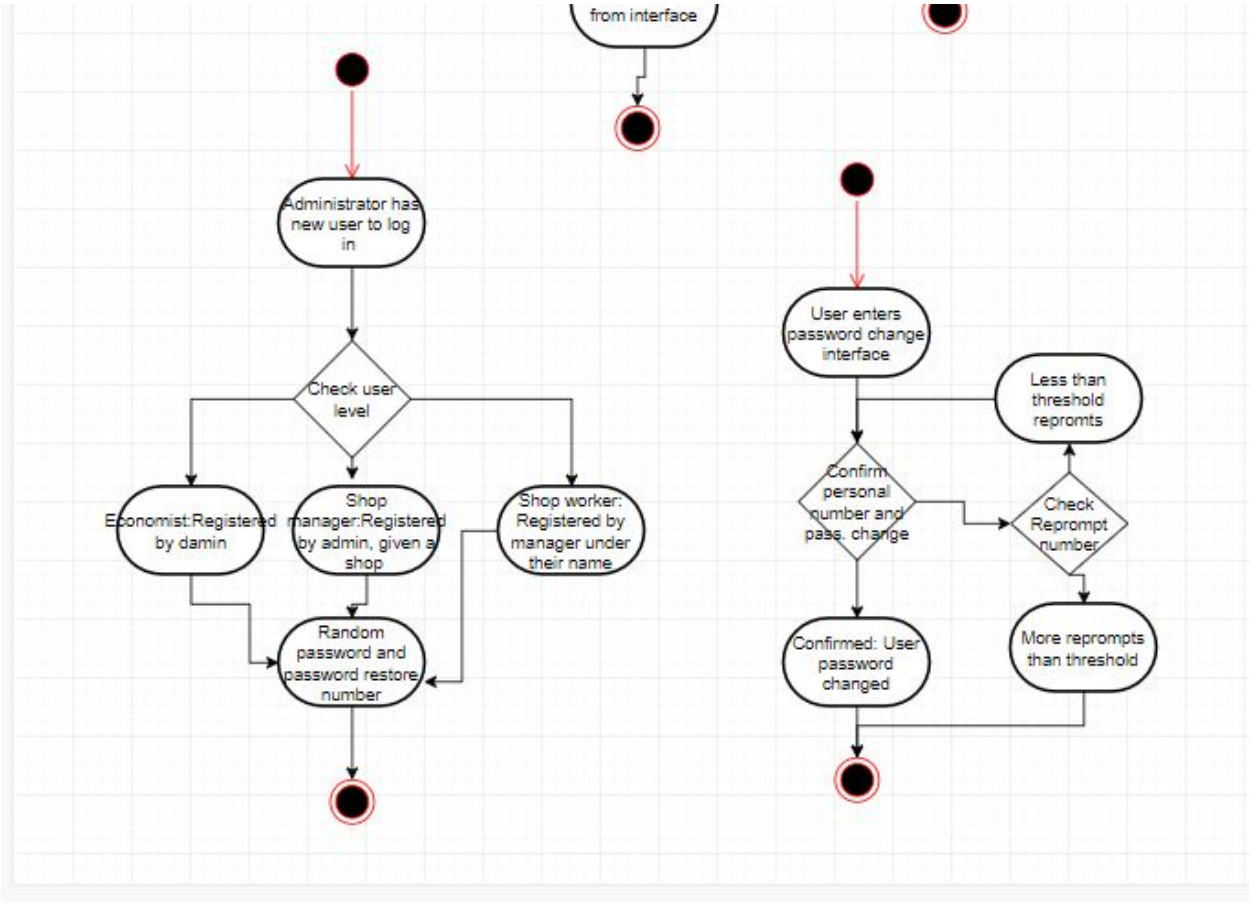
Cappuccino Use Case Diagram



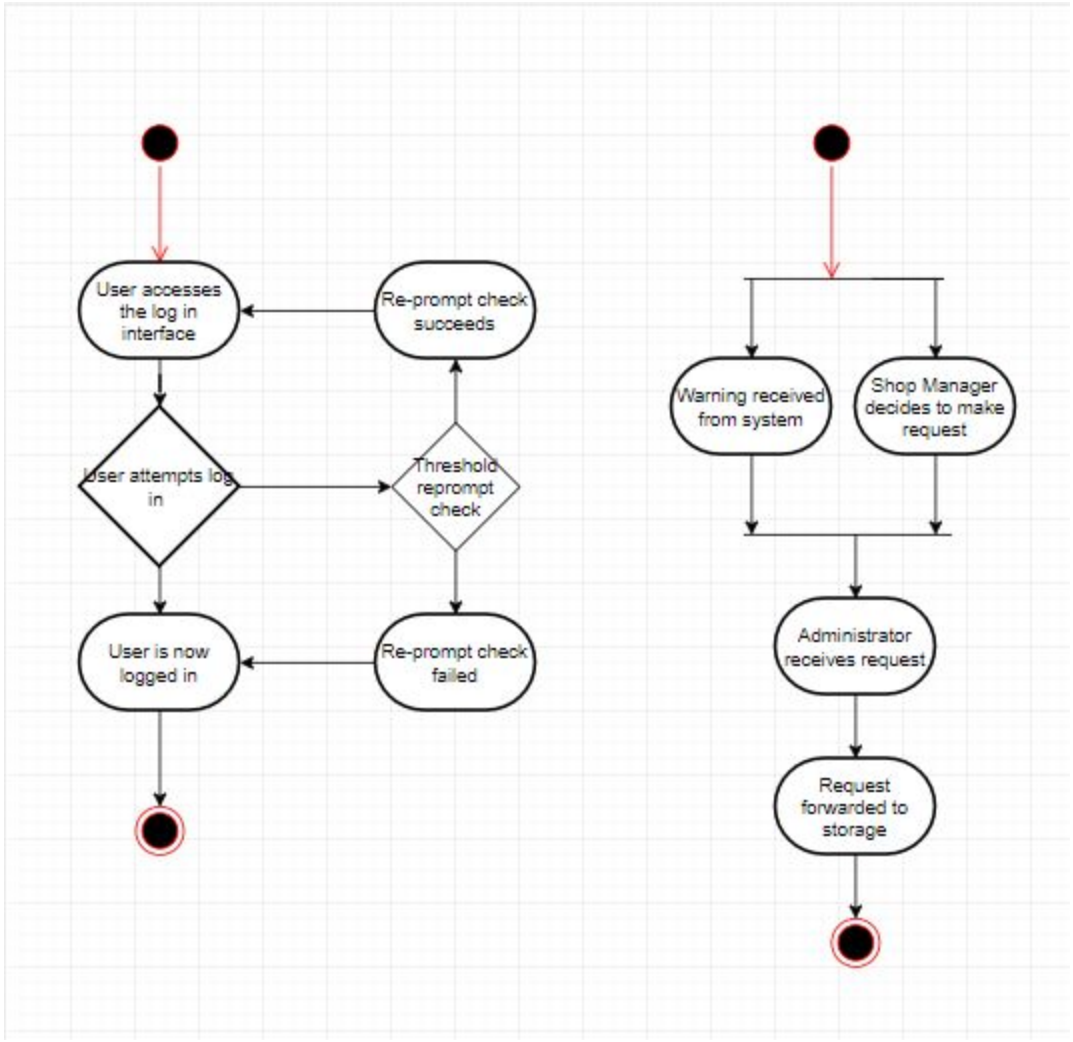
Cappuccino: Activity Diagrams



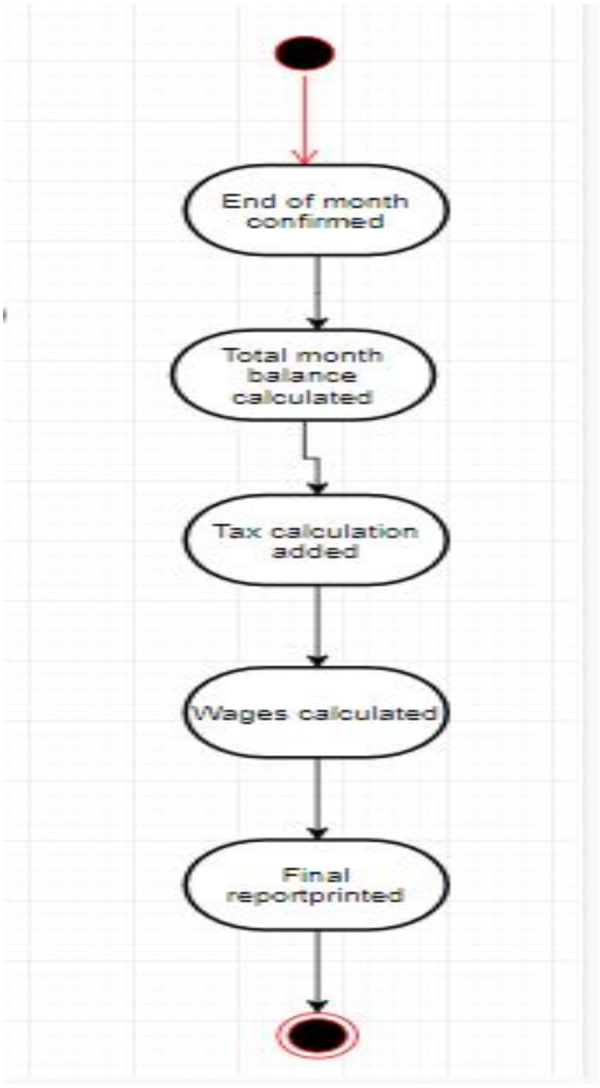
AC-01, AC-02, AC-03


from interface
Administrator has new user to log in
Check user level
Economist:Registered by damin
Shop manager:Registered by admin, given a shop
Shop worker: Registered by manager under their name
Random password and password restore number
User enters password change interface
Less than threshold reprmts
Confirm personal number and pass. change
Check Reprompt number
Confirmed: User password changed
More reprompts than threshold

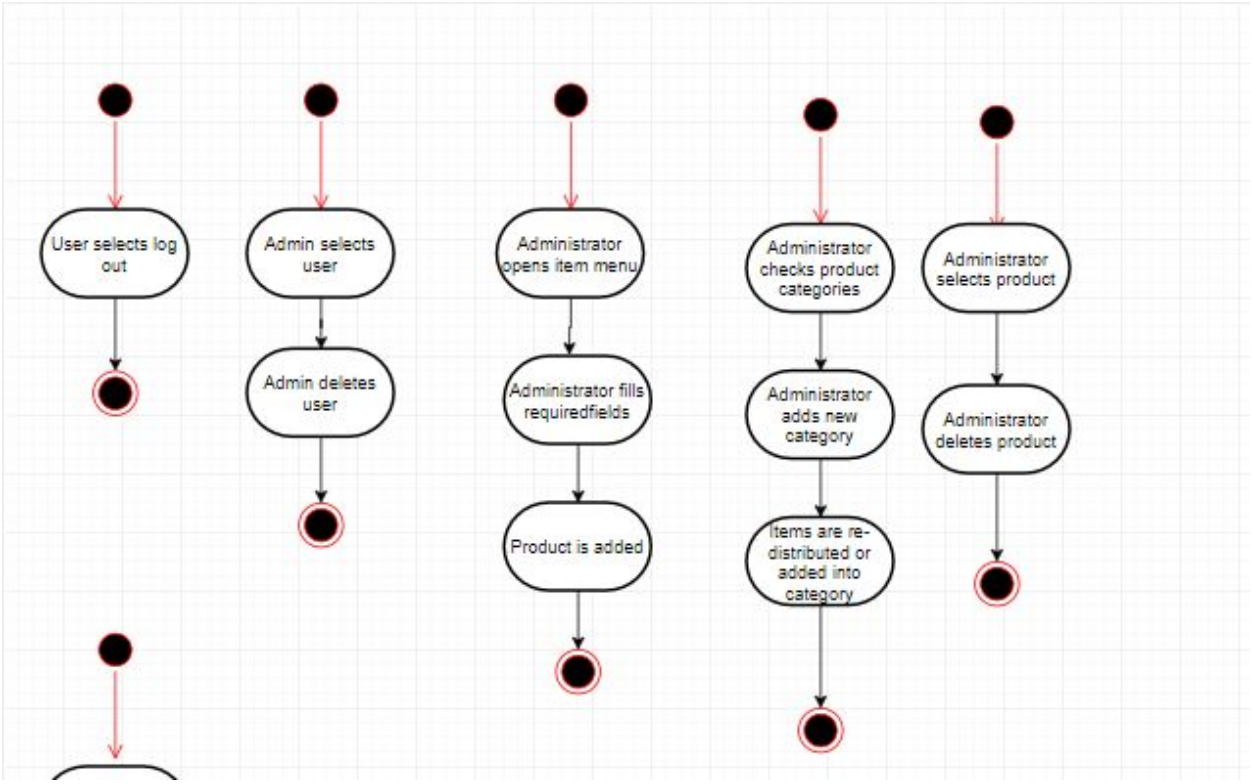
AC-04, AC-05

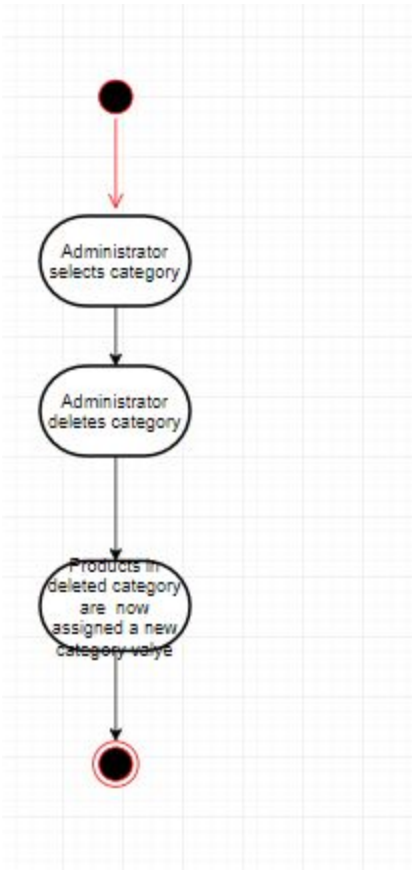

User accesses the log in interface
Re-prompt check succeeds
User attempts log in
Threshold reprompt check
User is now logged in
Re-prompt check failed
Warning received from system
Shop Manager decides to make request
Administrator receives request
Request forwarded to storage

AC-06, AC-07


End of month confirmed
Total month balance calculated
Tax calculation added
Wages calculated
Final reportprinted

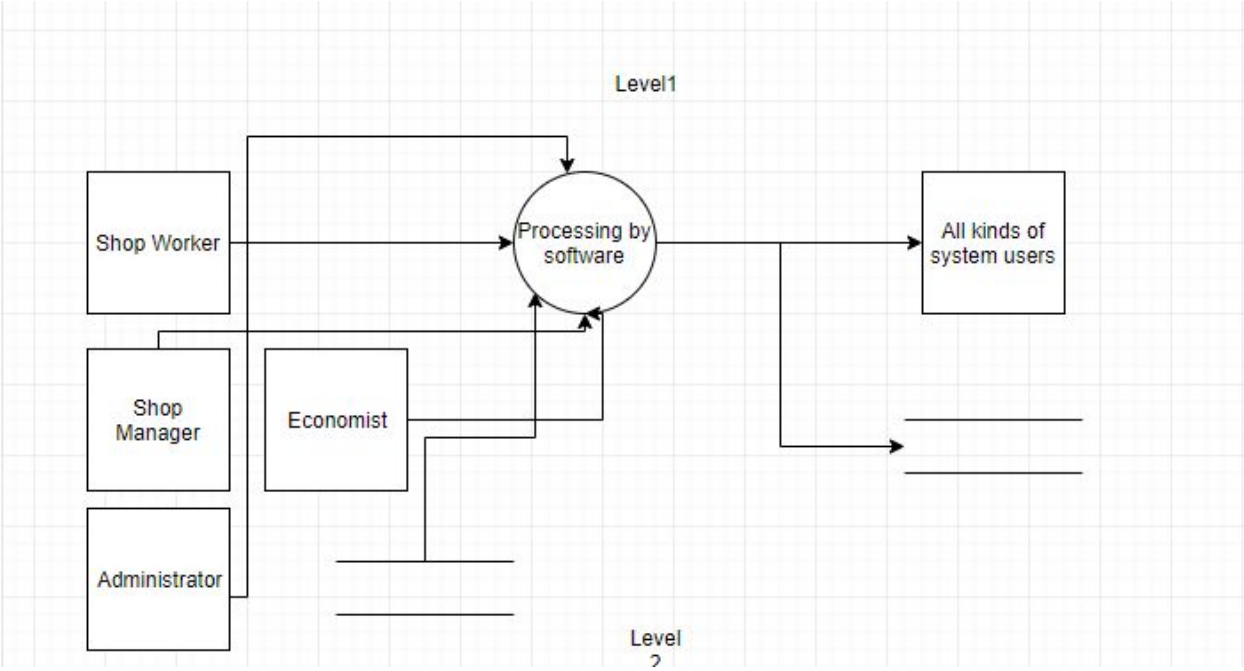
AC-08


User selects log out
Admin selects user
Admin deletes user
Administrator opens item menu
Administrator fills requiredfields
Product is added
Administrator checks product categories
Administrator adds new category
Items are re-distributed or added into category
Administrator selects product
Administrator deletes product

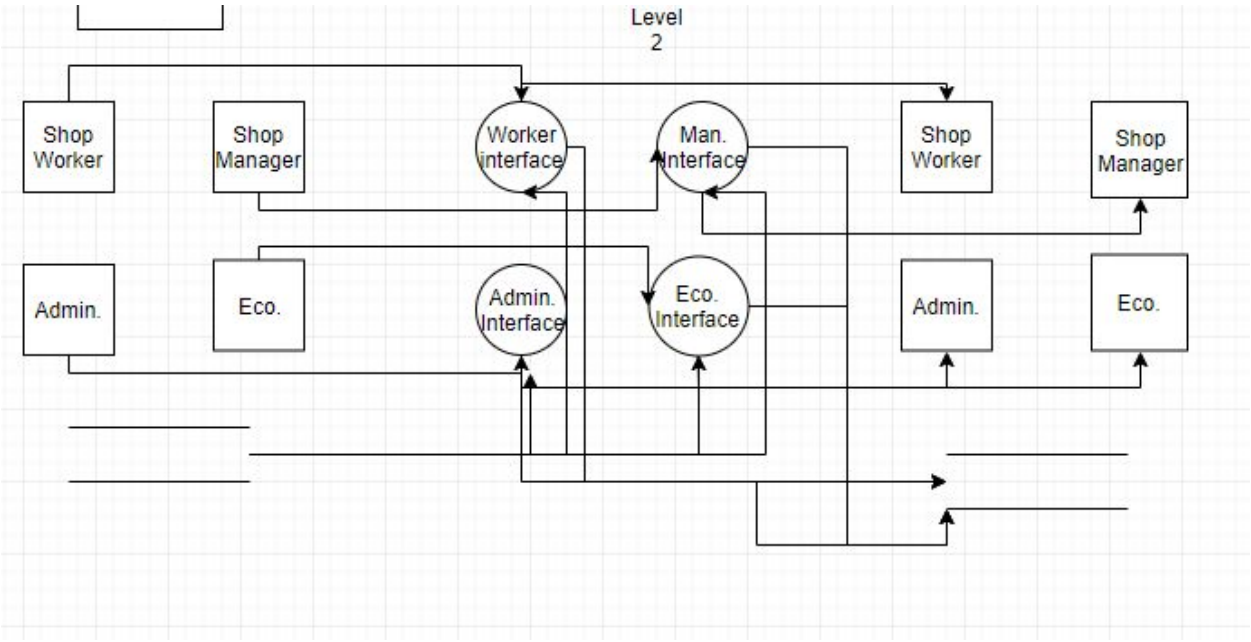

Administrator selects category
Administrator deletes category
Products in deleted category are now assigned a new category value

AC 09-14, AC-15 below

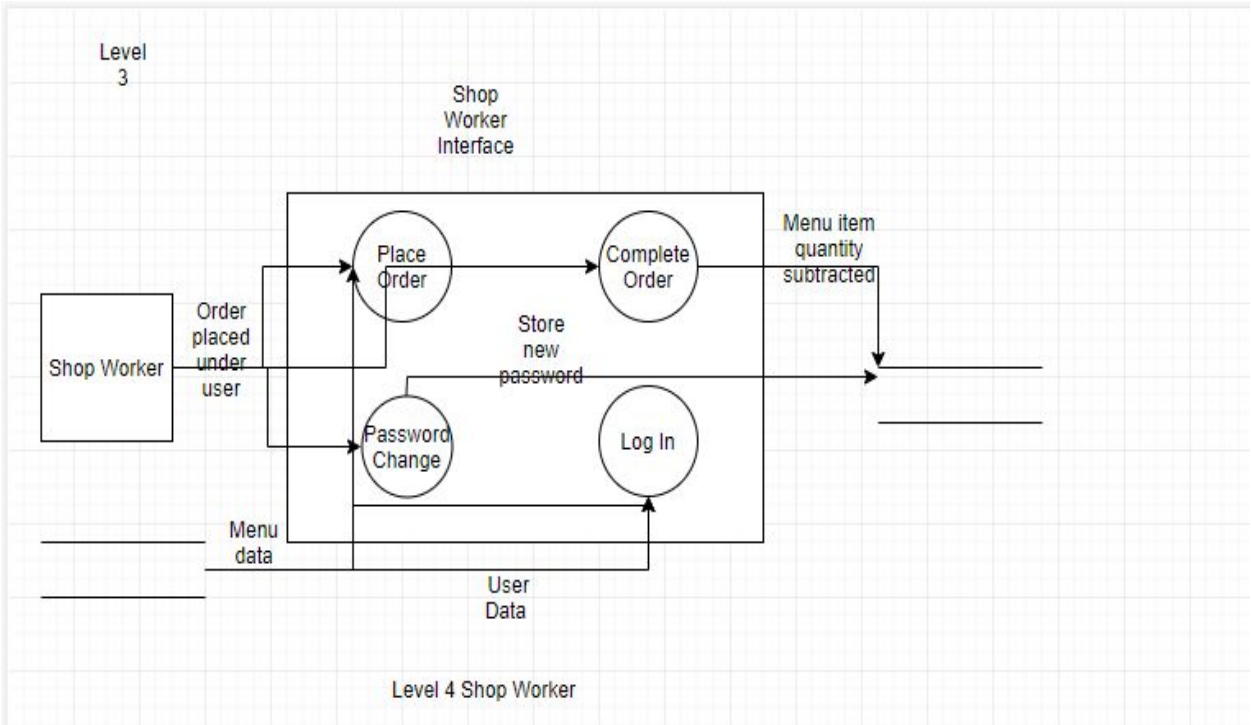
Cappuccino: DFD



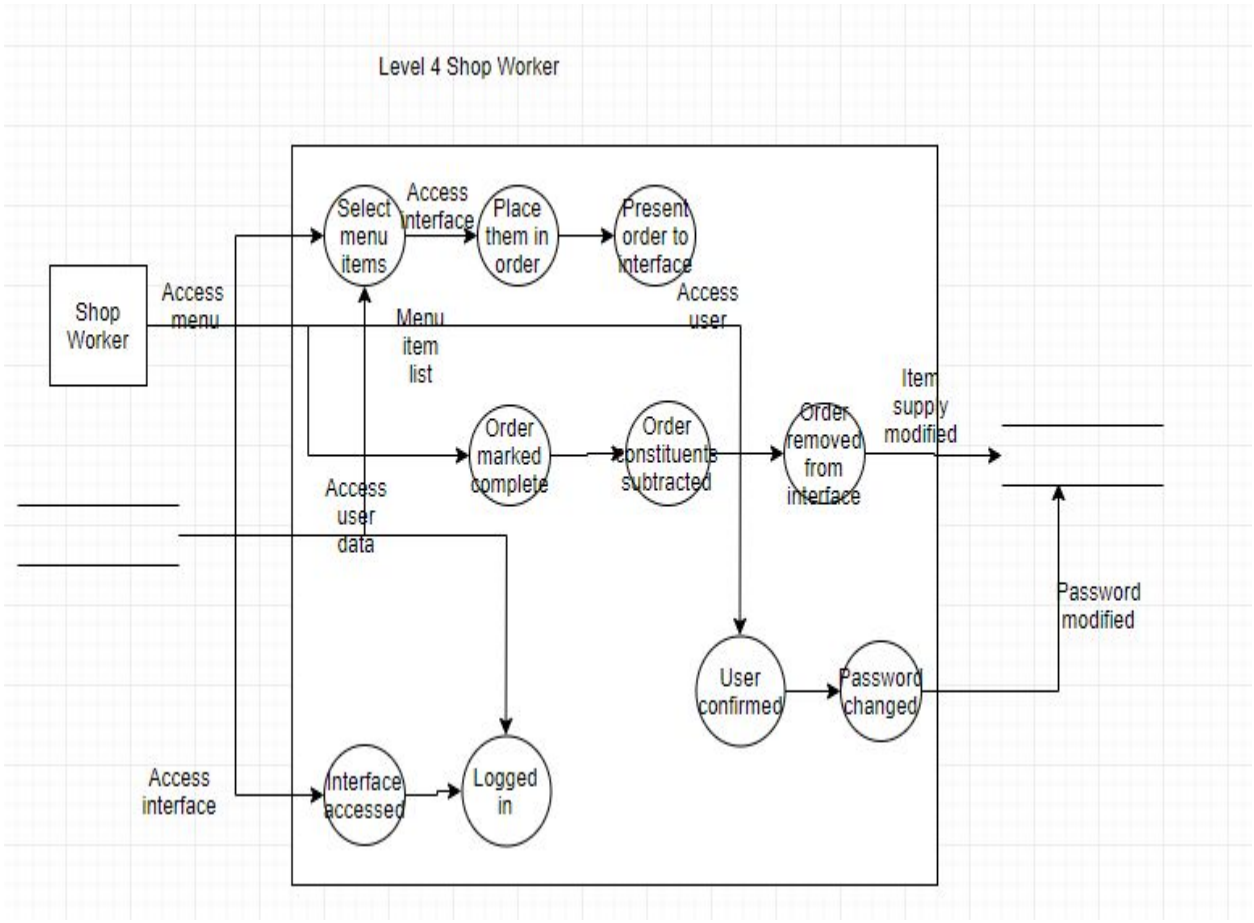
LEVEL 0 CAPPUCCINO DFD



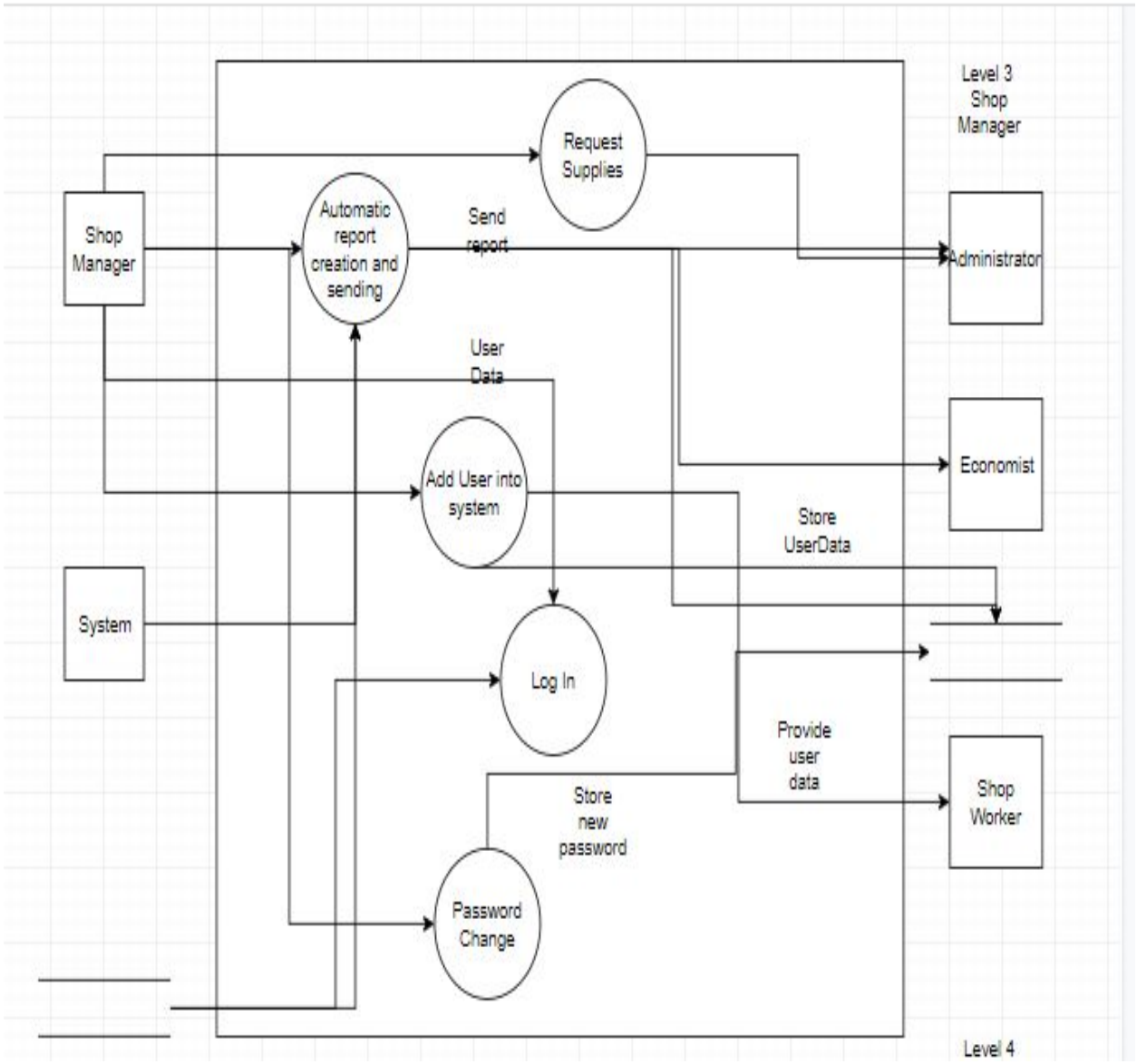
LEVEL 1 CAPPUCCINO DFD



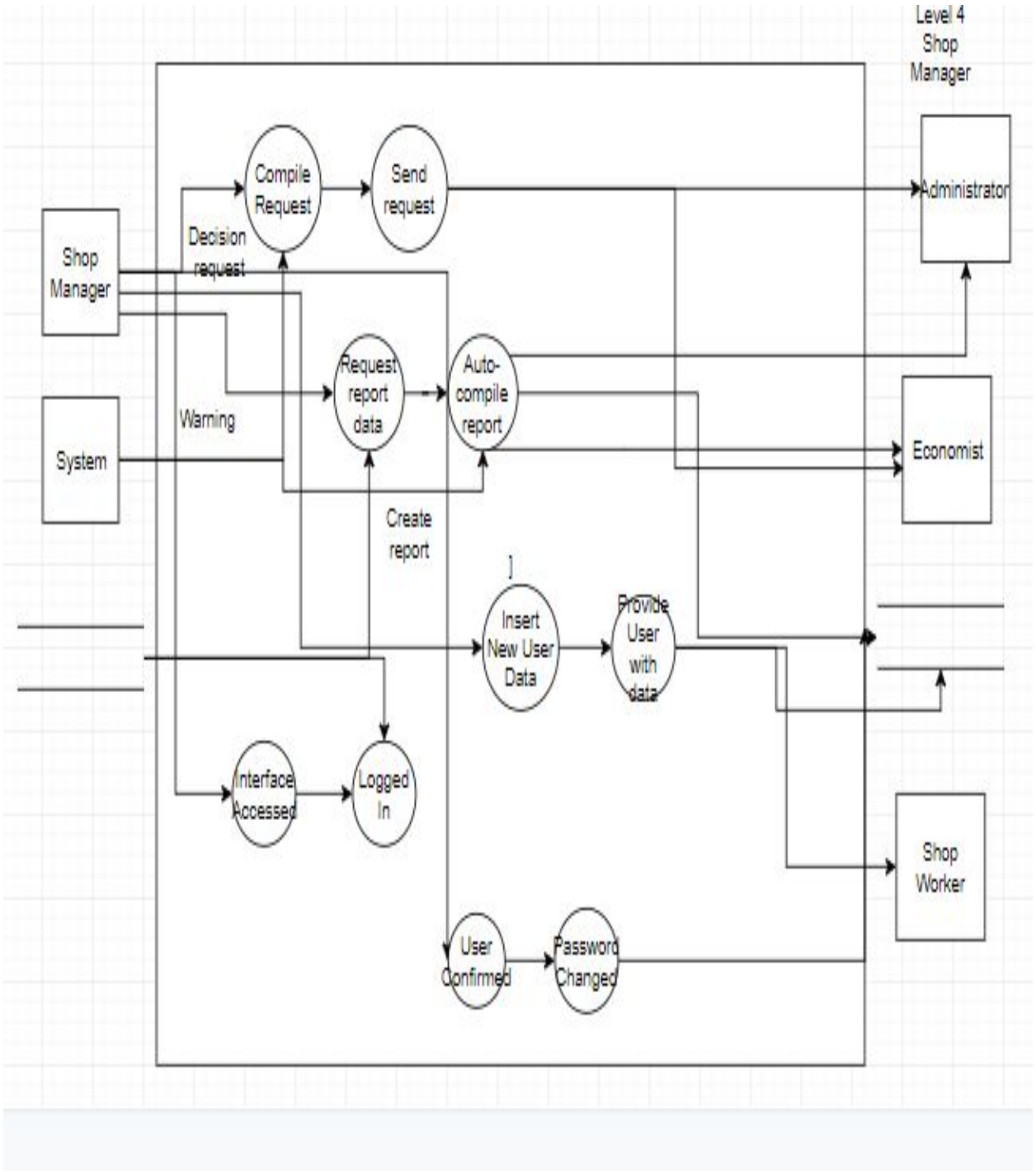
LEVEL 2 CAPUCCINO DFD, SHOP WORKER



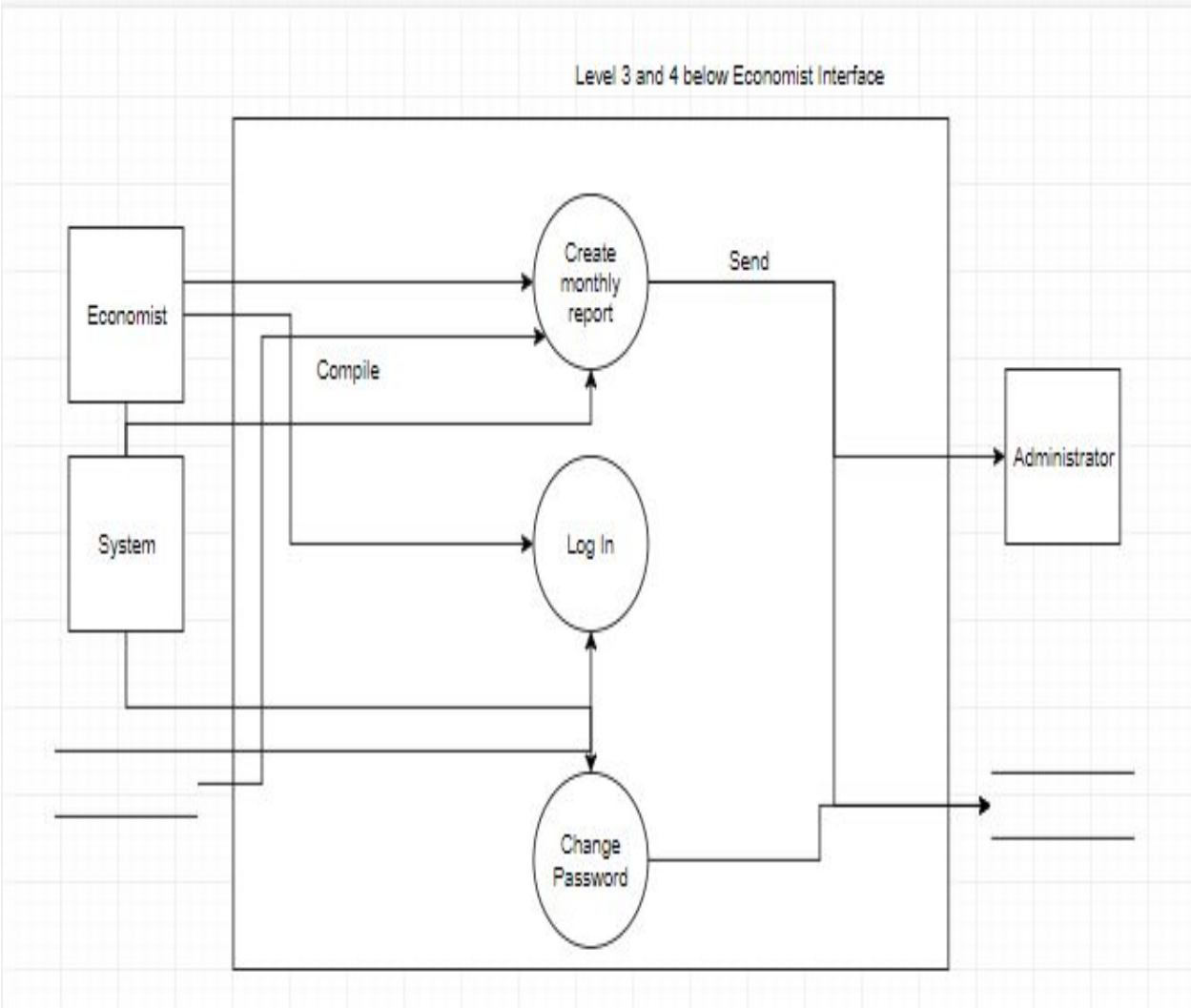
LEVEL 4 CAPPUCCINO DFD -SHOP WORKER



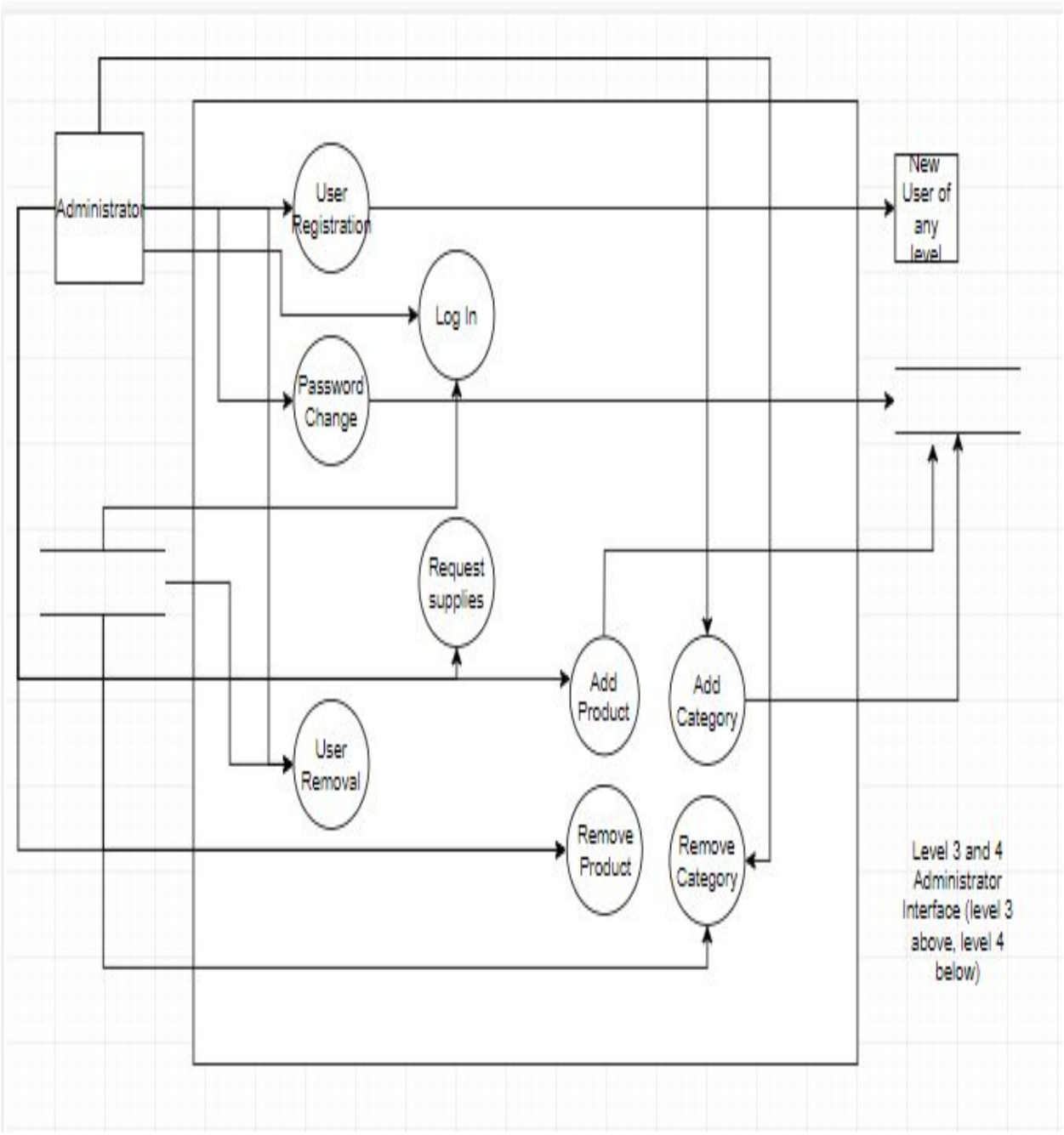
LEVEL 2 CAPPUCCINO DFD - SHOP MANAGER



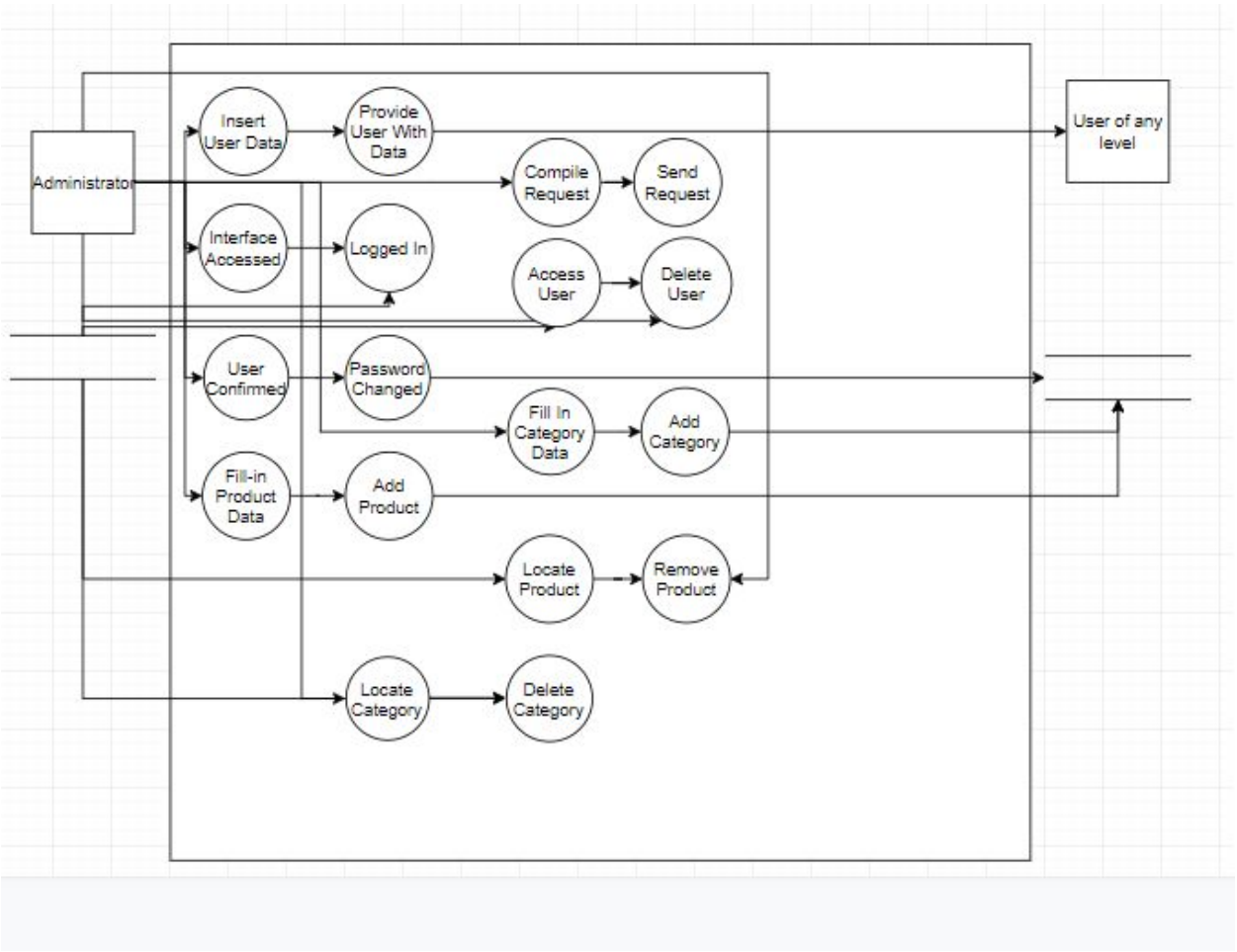
LEVEL 3 CAPPUCCINO DFD - SHOP MANAGER



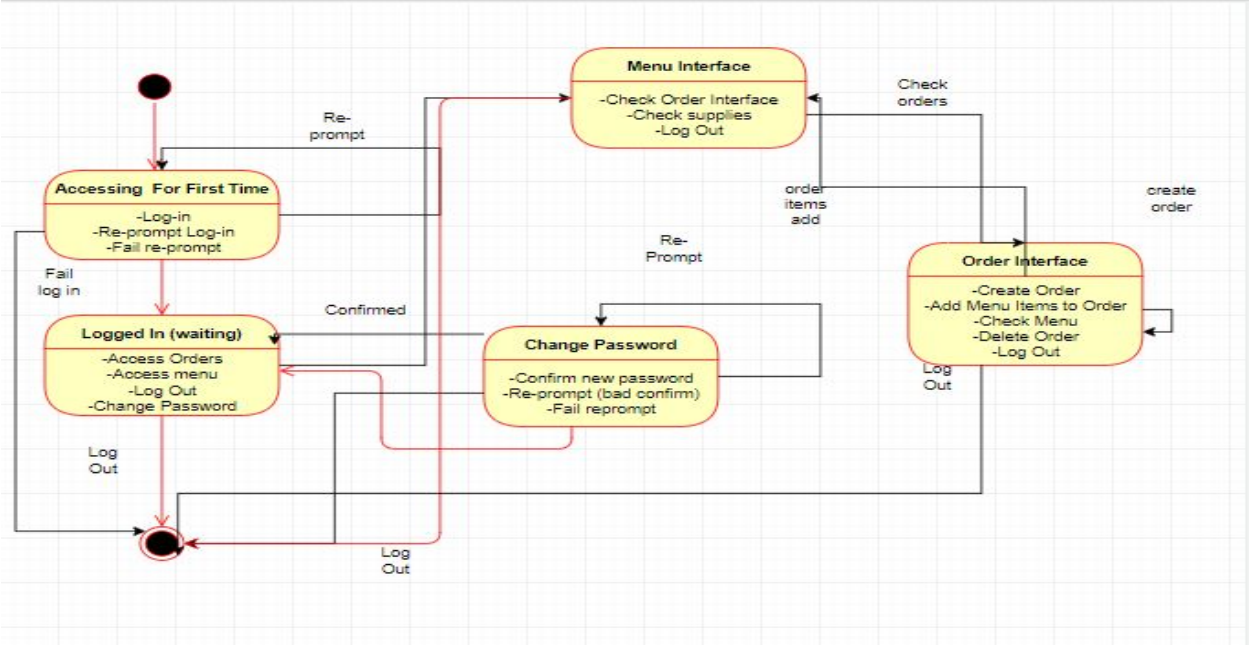
LEVEL 2 AND 3 DFD CAPPUCCINO, ECONOMIST



LEVEL 2 CAPPUCCINO DFD, ADMINISTRATOR

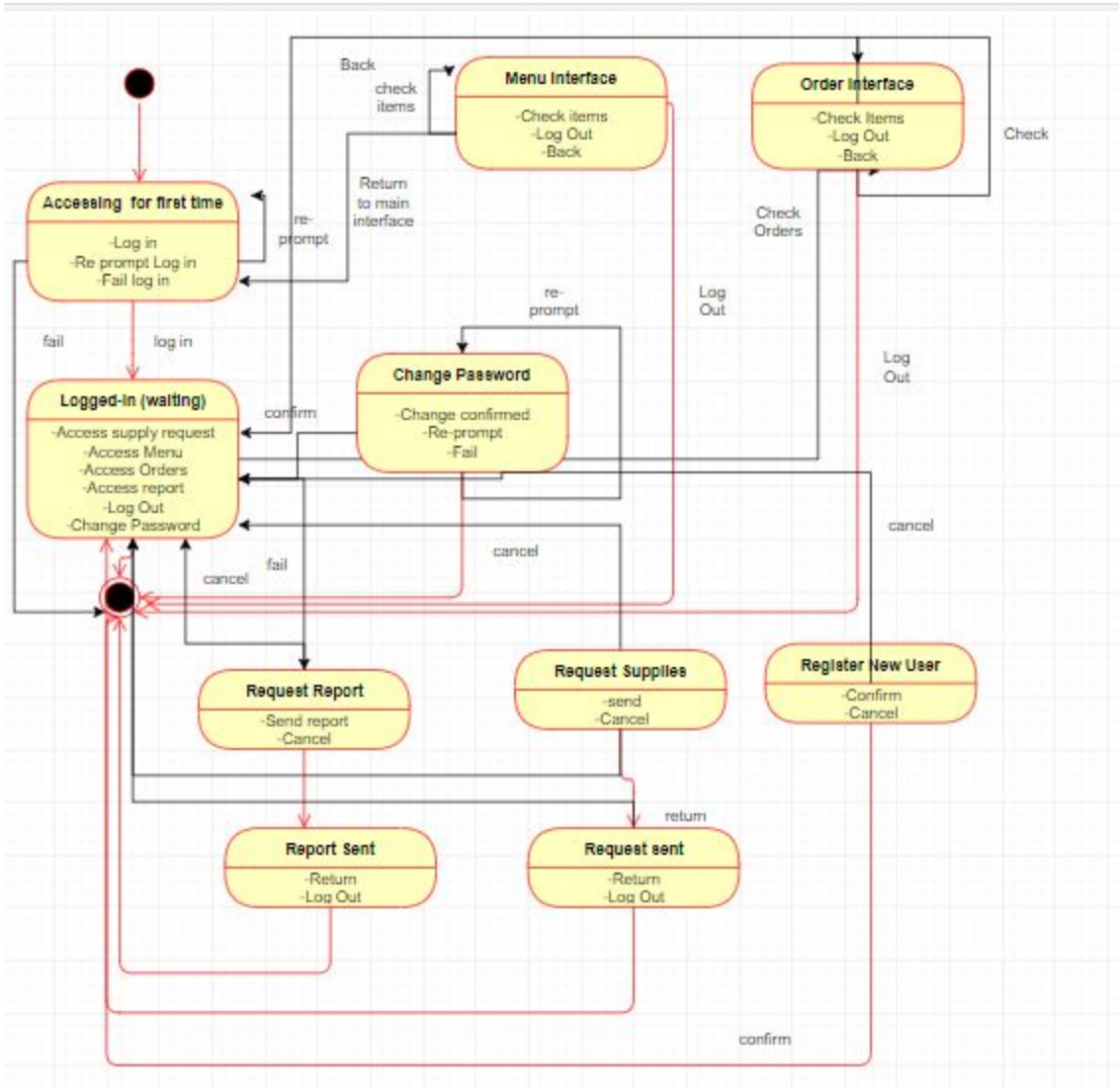


LEVEL 3 DFD CAPPUCCINO ADMINISTRATOR

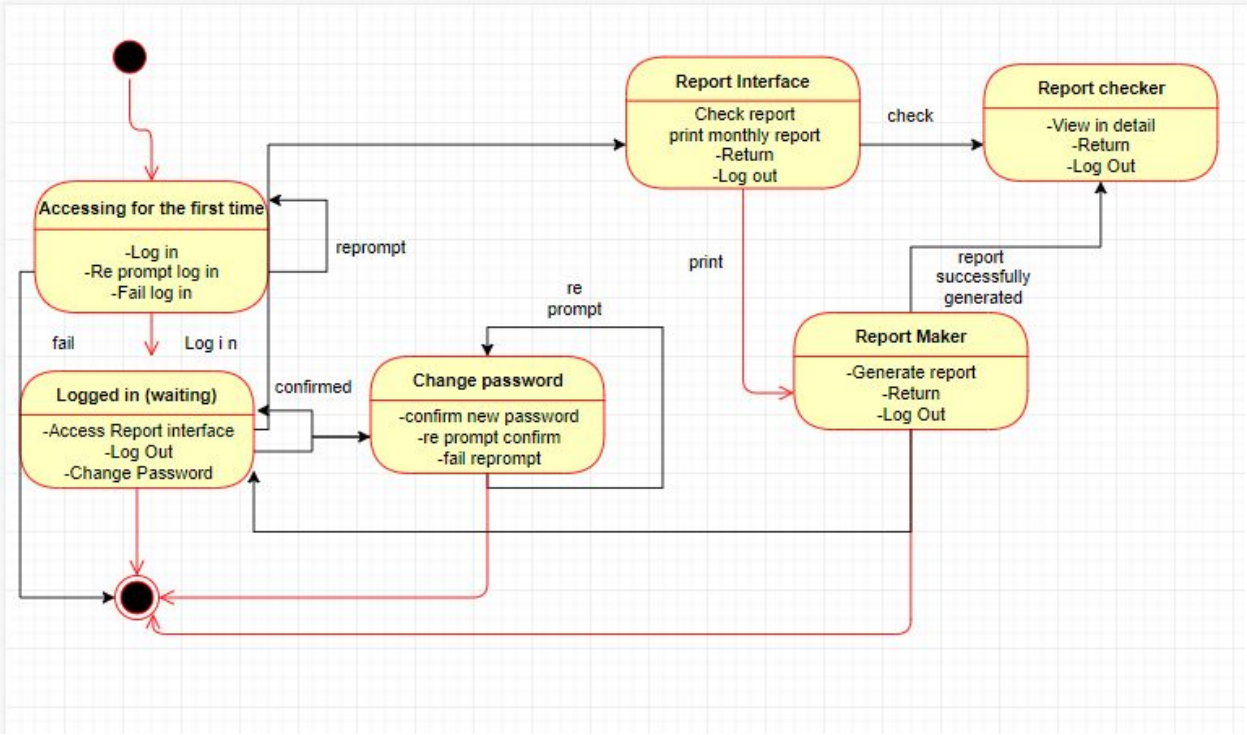


ST-01 - SHOP WORKER

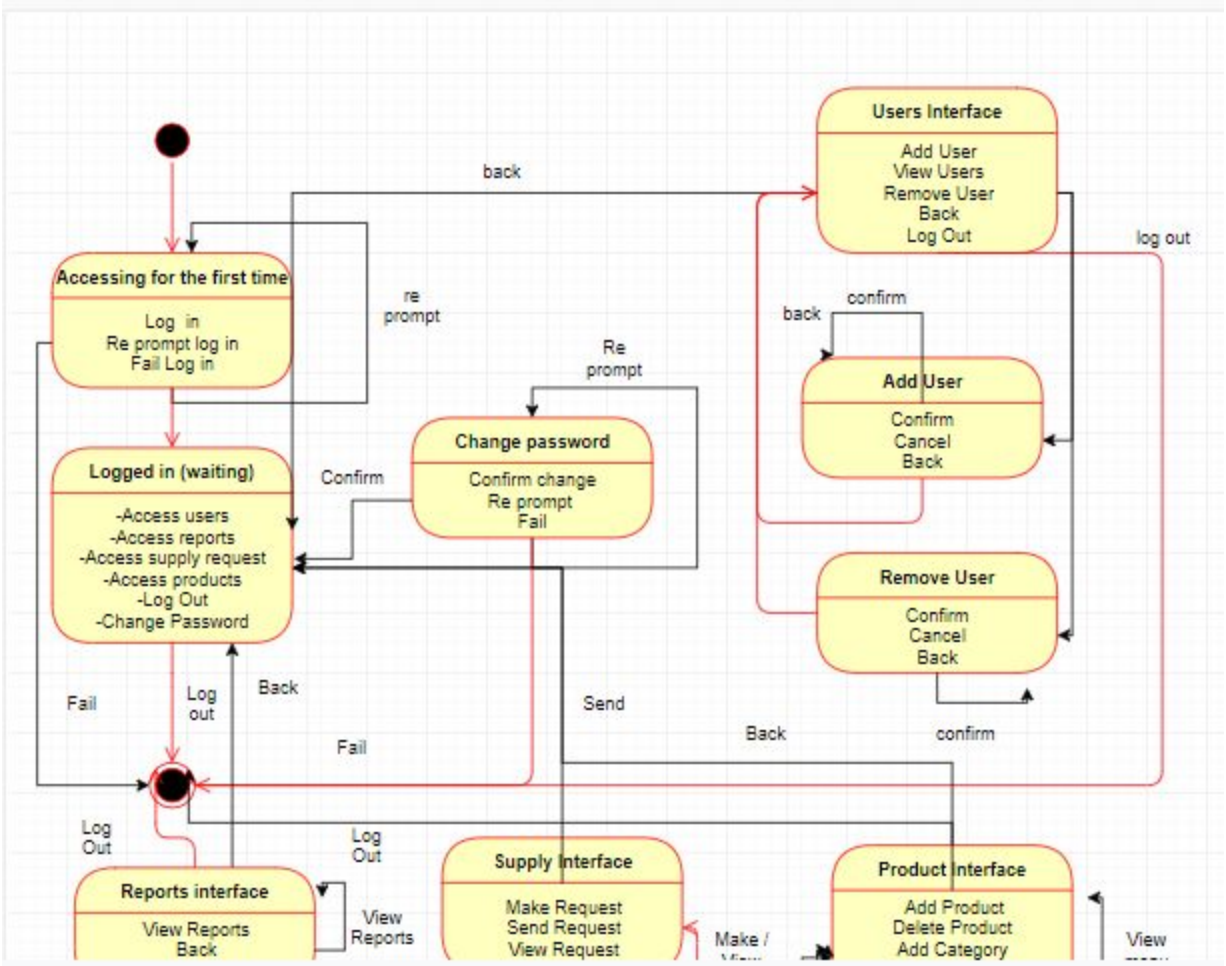
Cappuccino: State diagrams



ST-02 - SHOP MANAGER

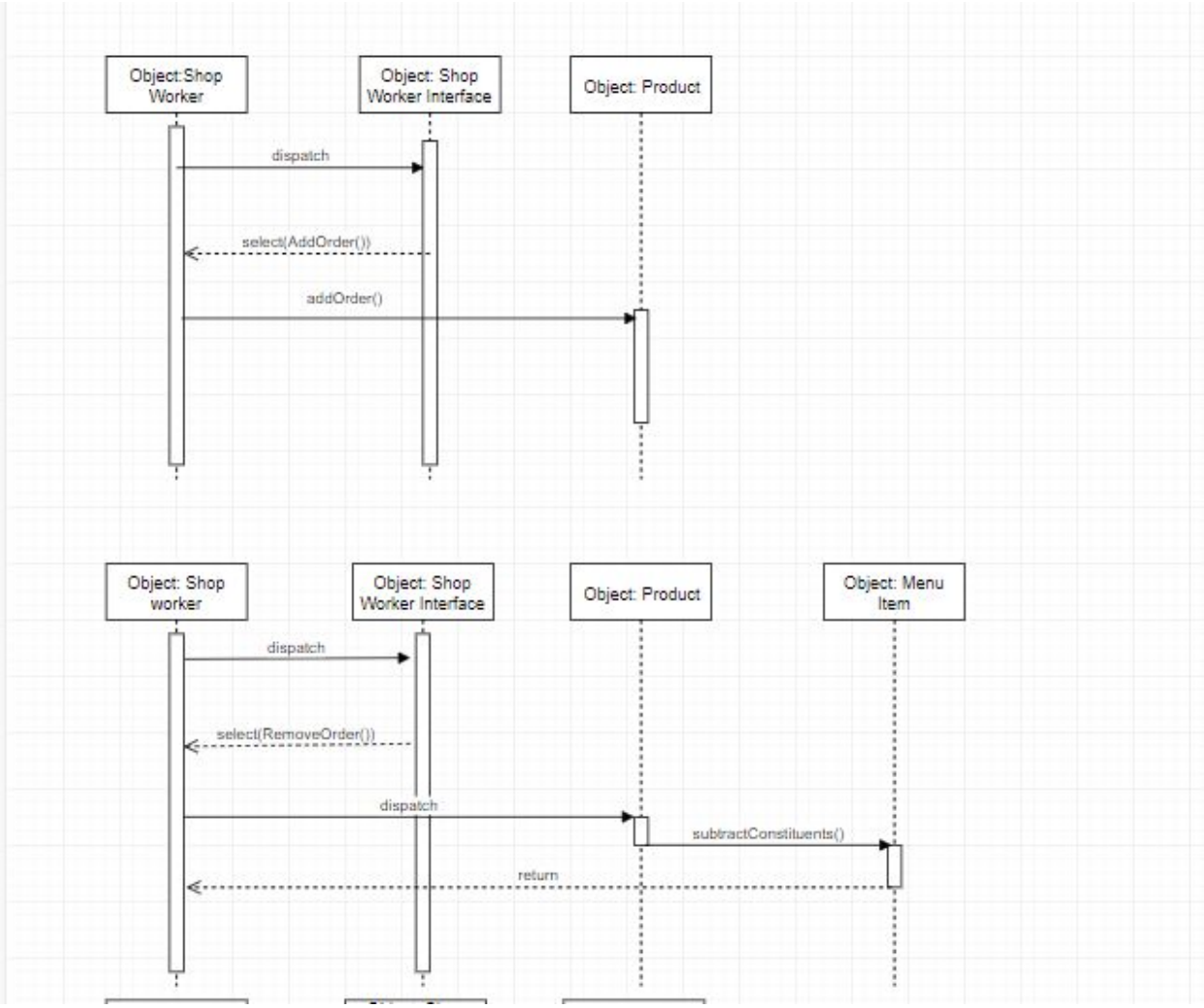


ST-03 - ECONOMIST

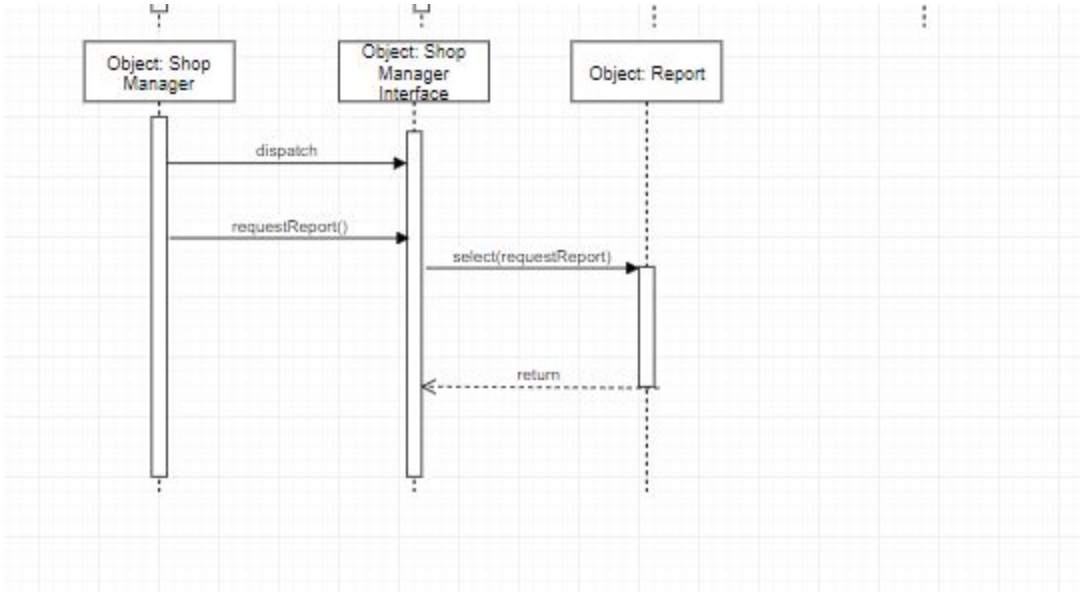


ST-04 - ADMINISTRATOR

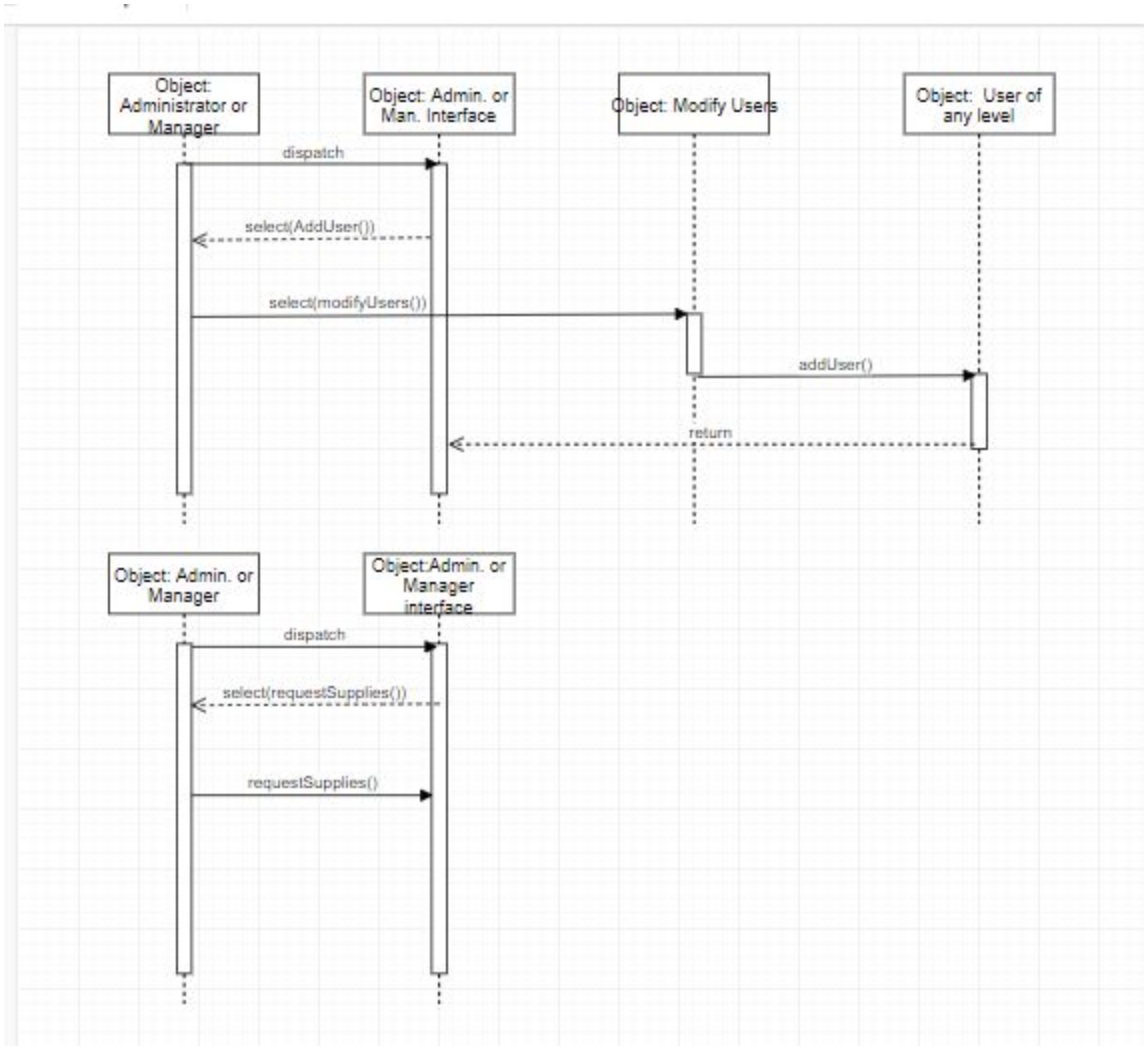
Cappuccino: Sequence Diagrams


Object:Shop Worker
Object: Shop Worker Interface
Object: Product
dispatch
select(AddOrder())
addOrder()
Object: Shop worker
Object: Shop Worker Interface
Object: Product
Object: Menu Item
dispatch
select(RemoveOrder())
dispatch
subtractConstituents()
return

SEQ 01
SEQ 02

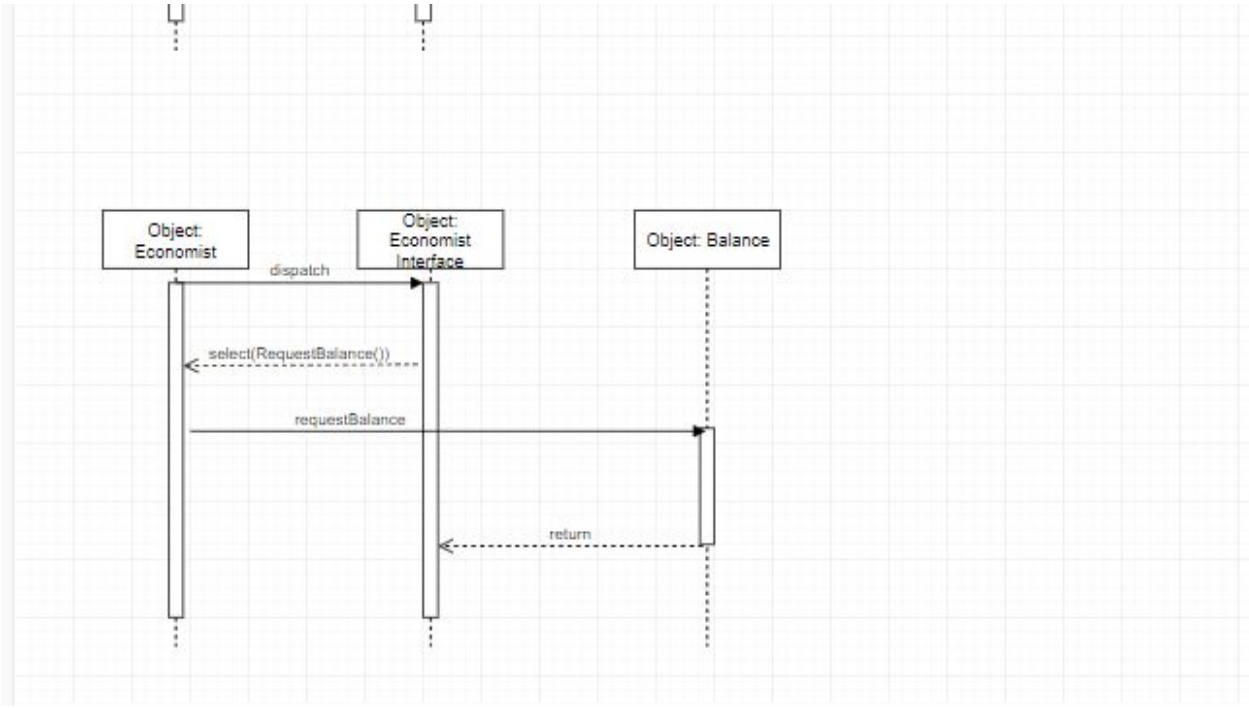

Object: Shop Manager
Object: Shop Manager Interface
Object: Report
dispatch
requestReport()
select(requestReport)
return

SEQ -03

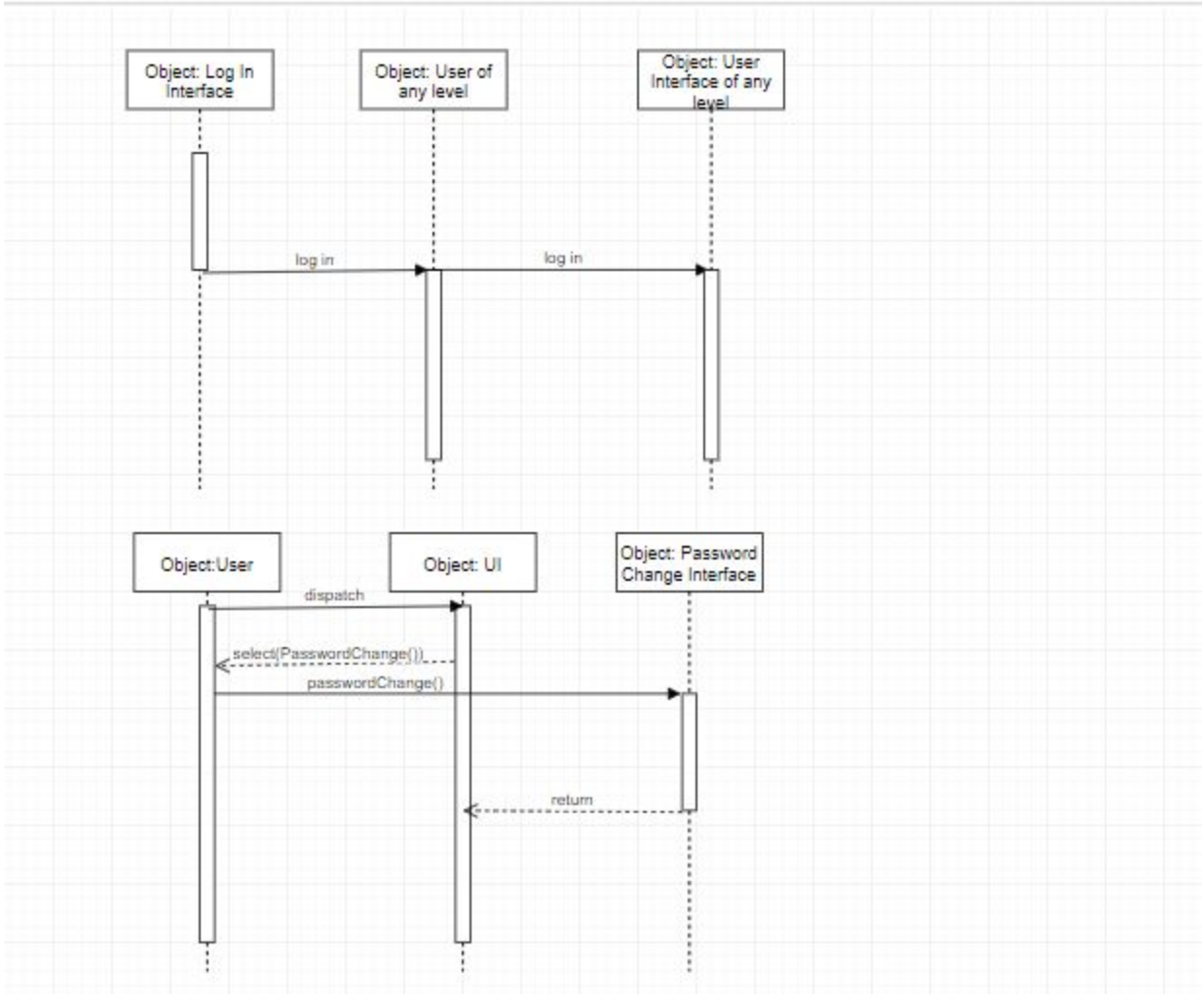

Object: Administrator or Manager
Object: Admin. or Man. Interface
Object: Modify Users
Object: User of any level
dispatch
select(AddUser())
select(modifyUsers())
addUser()
return
Object: Admin. or Manager
Object:Admin. or Manager interface
dispatch
select(requestSupplies())
requestSupplies()

SEQ 04

SEQ 05

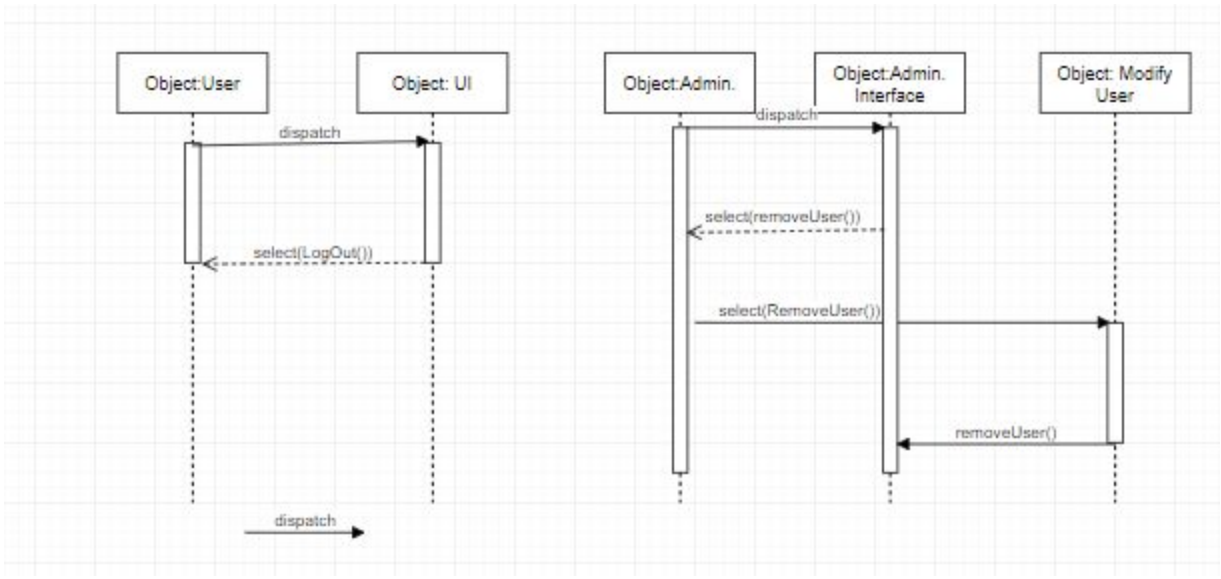

Object:
Economist
Object:
Economist
Interface
Object: Balance
dispatch
select(RequestBalance())
requestBalance
return

SEQ 06

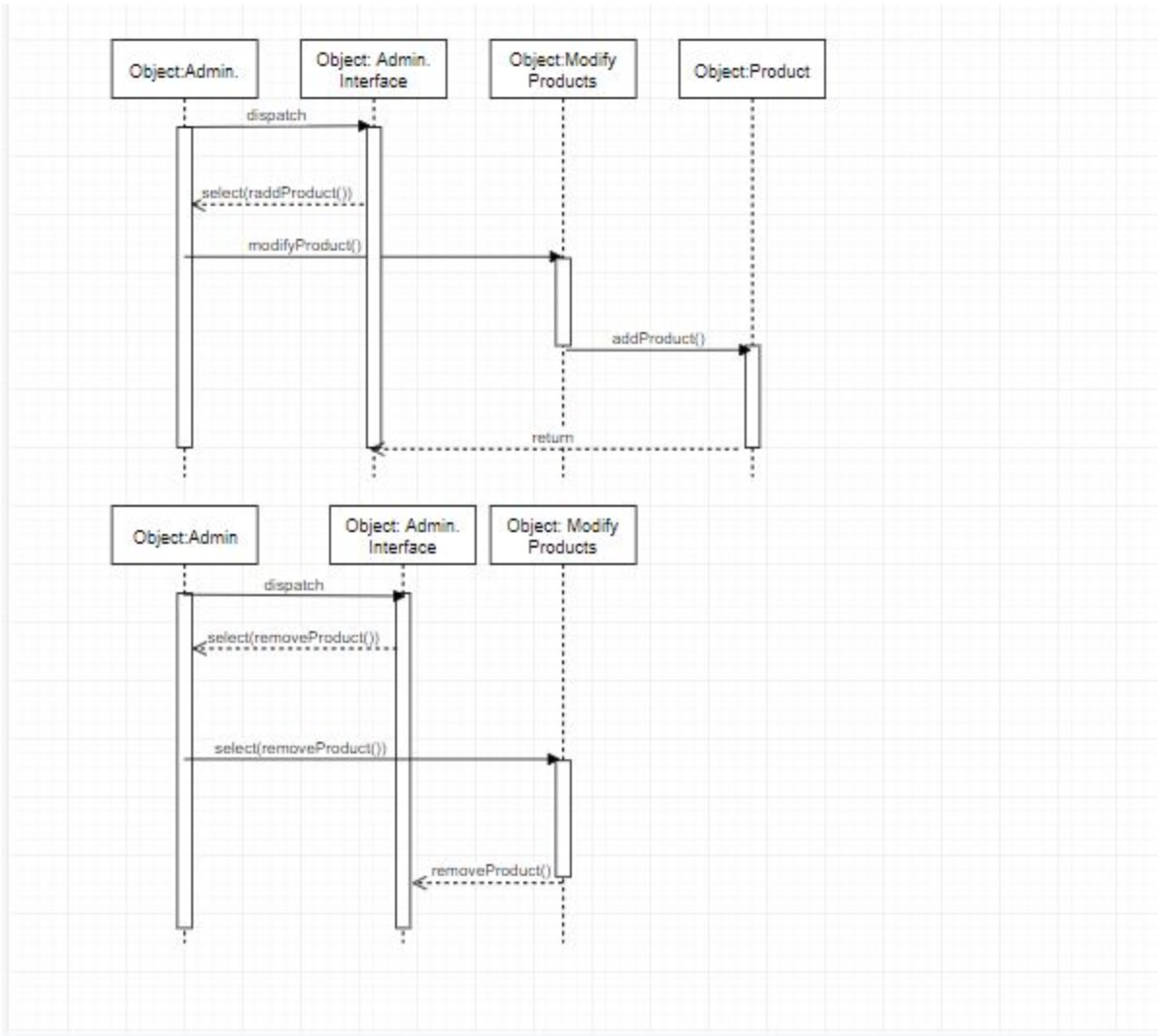

Object: Log In Interface
Object: User of any level
Object: User Interface of any level
log in
log in
Object:User
Object: UI
Object: Password Change Interface
dispatch
select(PasswordChange())
passwordChange()
return

SEQ 07

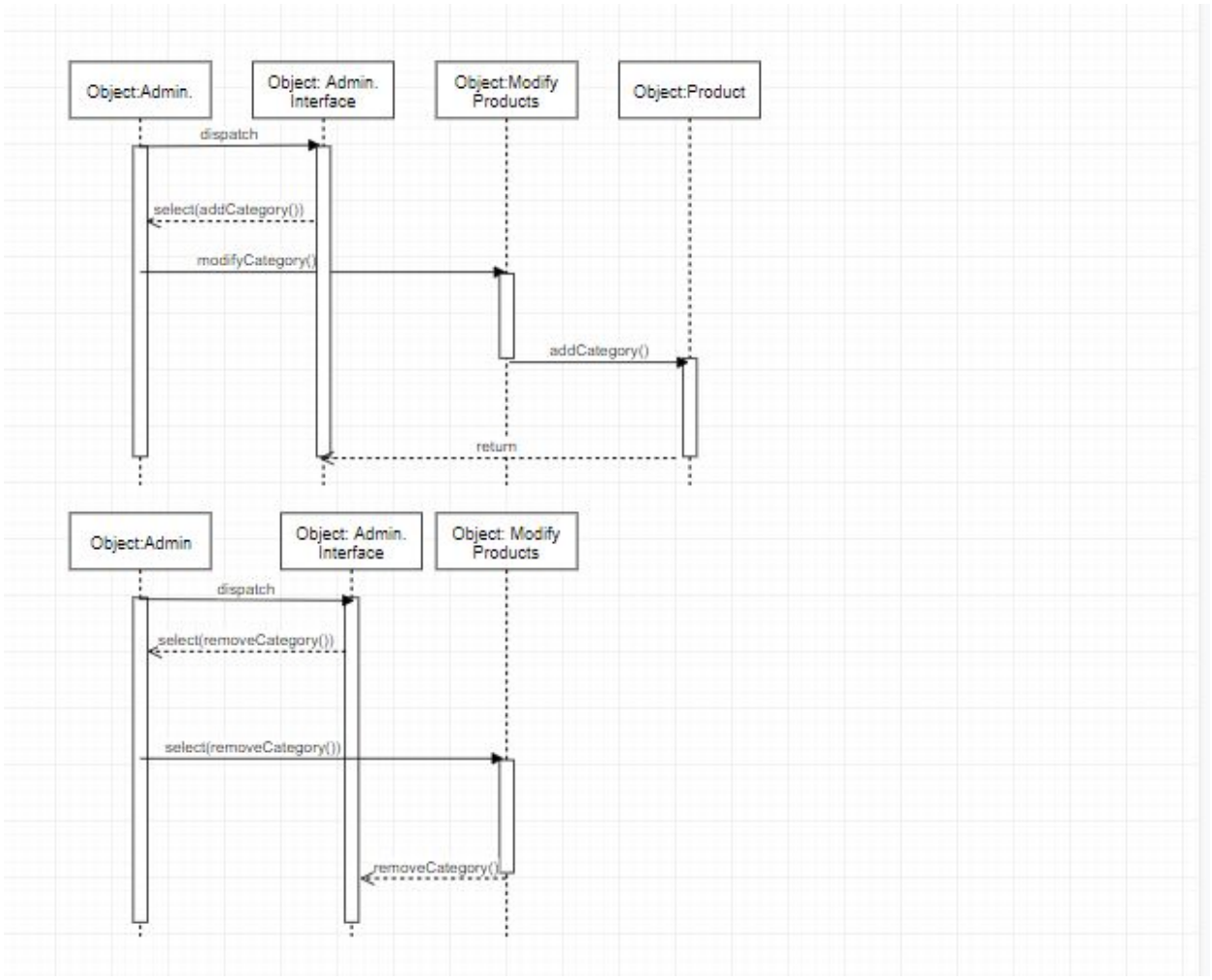
SEQ 08


Object:User
Object: UI
dispatch
select(LogOut())
dispatch
Object:Admin.
Object:Admin. Interface
Object: Modify User
dispatch
select(removeUser())
select(RemoveUser())
removeUser()

SEQ 09 , SEQ 10

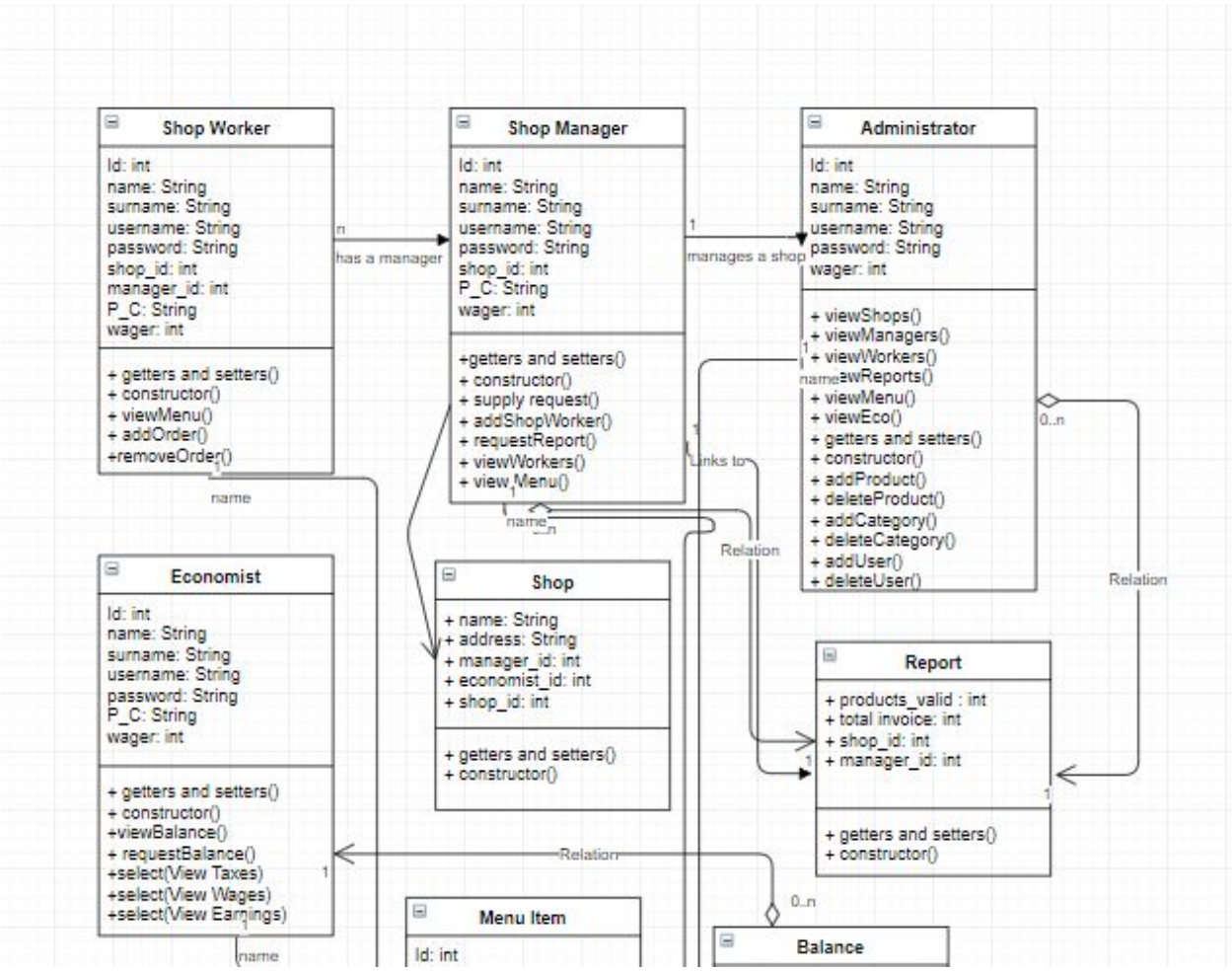

Object:Admin.
Object: Admin. Interface
Object:Modify Products
Object:Product
dispatch
select(raddProduct())
modifyProduct()
addProduct()
return
Object:Admin
Object: Admin. Interface
Object: Modify Products
dispatch
select(removeProduct())
select(removeProduct())
removeProduct()

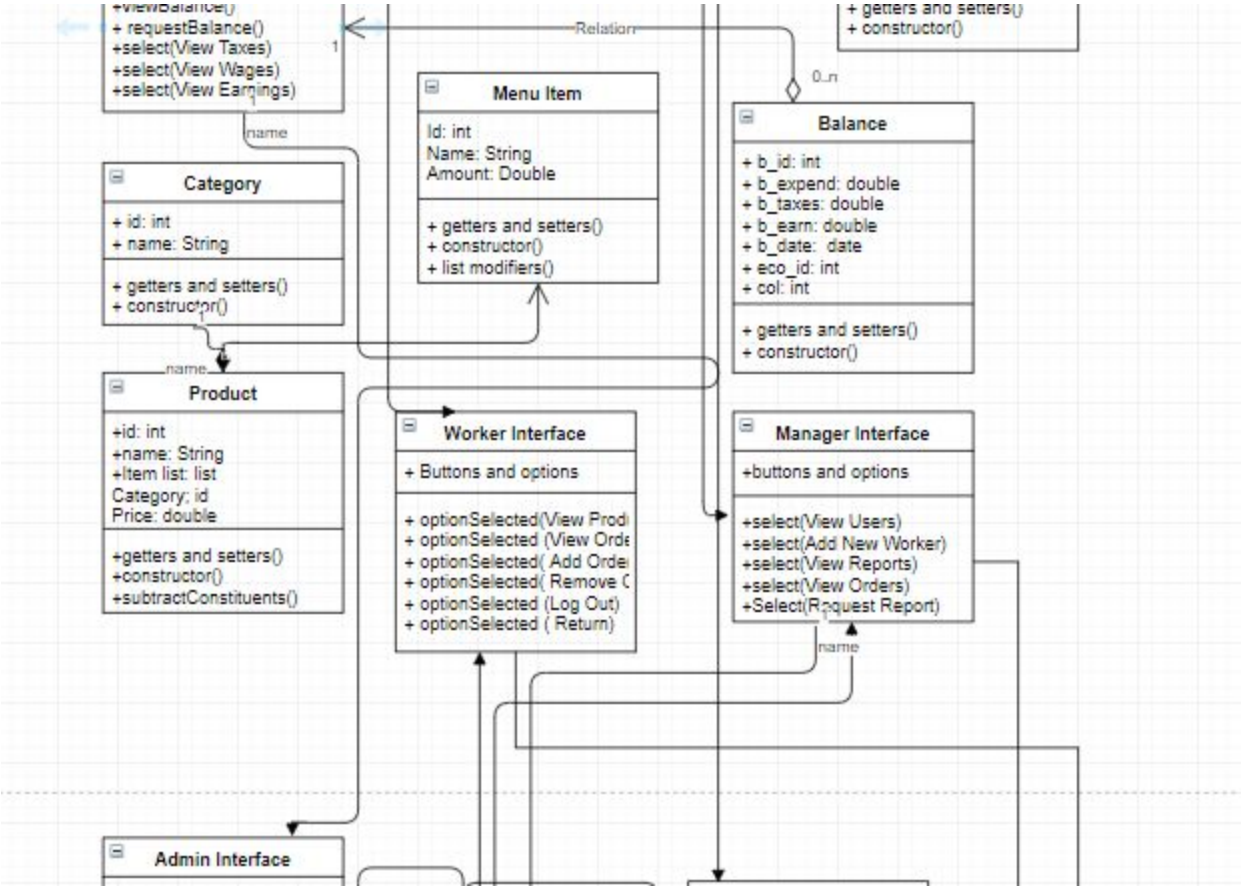
SEQ 11, SEQ 12

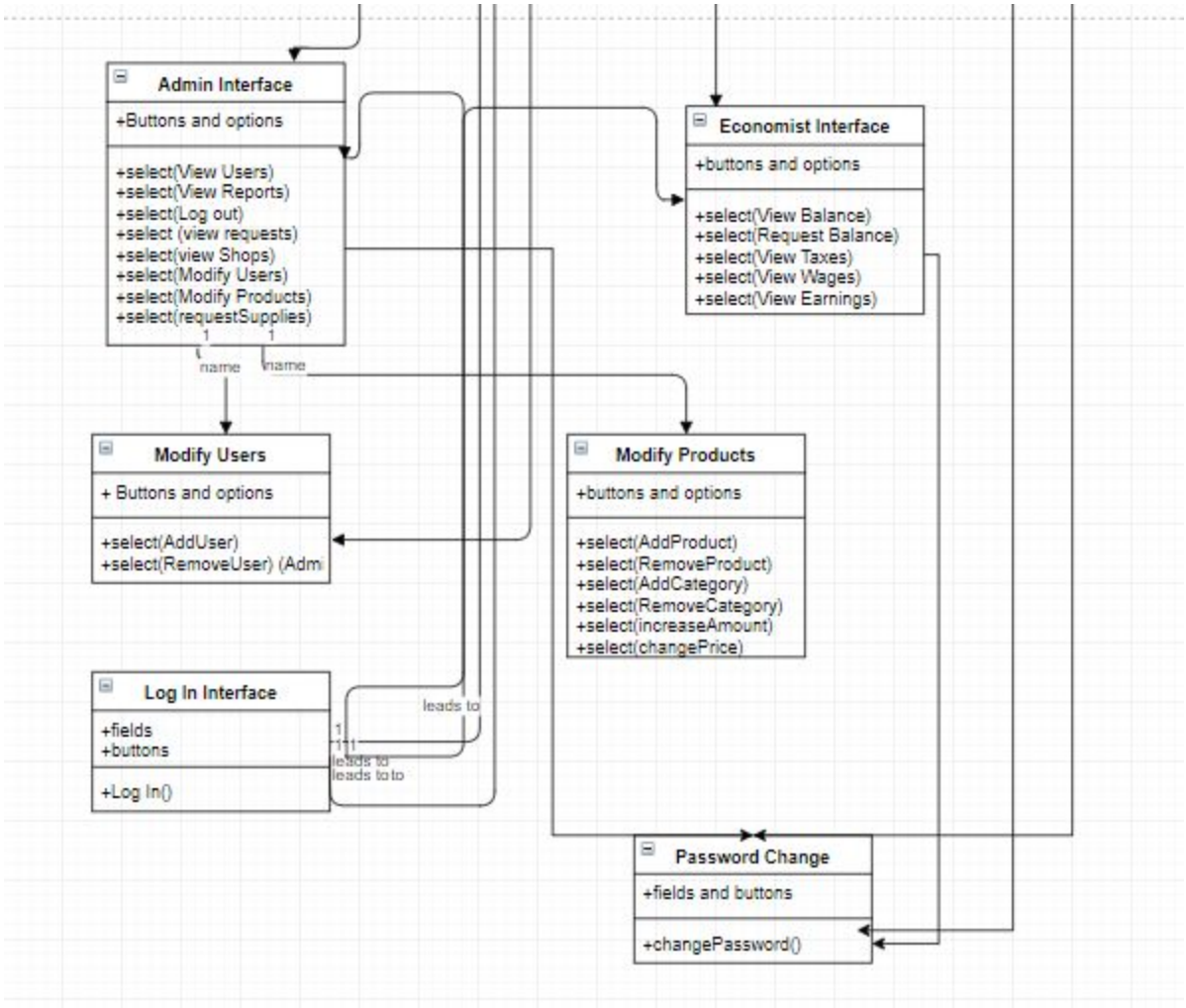


SEQ 13, SEQ 14

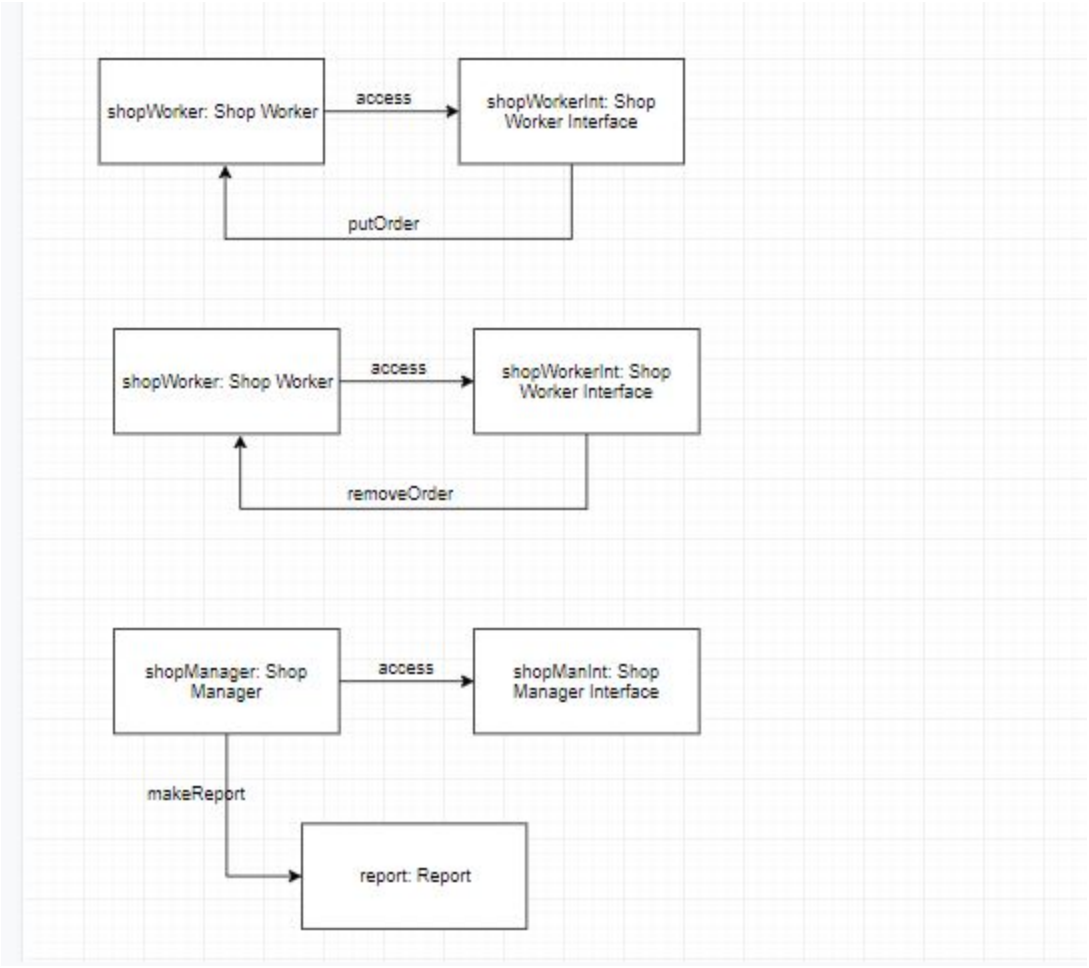
Cappuccino: Class Diagram


Shop Worker
Id: int
name: String
surname: String
username: String
password: String
shop_id: int
manager_id: int
P_C: String
wager: int
+ getters and setters()
+ constructor()
+ viewMenu()
+ addOrder()
+removeOrder()
n
has a manager
Shop Manager
Id: int
name: String
surname: String
username: String
password: String
shop_id: int
P_C: String
wager: int
+getters and setters()
+ constructor()
+ supply request()
+ addShopWorker()
+ requestReport()
+ viewWorkers()
+ view Menu()
1
manages a shop
Administrator
Id: int
name: String
surname: String
username: String
password: String
wager: int
+ viewShops()
+ viewManagers()
+ viewWorkers()
+ viewReports()
+ viewMenu()
+ viewEco()
+ getters and setters()
+ constructor()
+ addProduct()
+ deleteProduct()
+ addCategory()
+ deleteCategory()
+ addUser()
+ deleteUser()
0..n
Relation
Links to
name
Economist
Id: int
name: String
surname: String
username: String
password: String
P_C: String
wager: int
+ getters and setters()
+ constructor()
+viewBalance()
+ requestBalance()
+select(View Taxes)
+select(View Wages)
+select(View Earnings)
Shop
+ name: String
+ address: String
+ manager_id: int
+ economist_id: int
+ shop_id: int
+ getters and setters()
+ constructor()
Report
+ products_valid : int
+ total invoice: int
+ shop_id: int
+ manager_id: int
+ getters and setters()
+ constructor()
Relation
Menu Item
Id: int
Balance
0..n

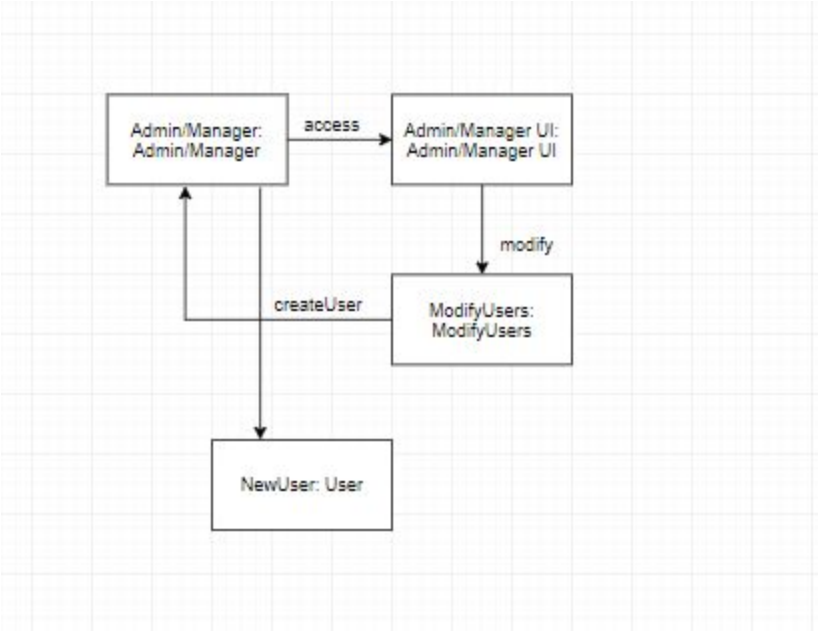

+viewBalance()
+ requestBalance()
+select(View Taxes)
+select(View Wages)
+select(View Earnings)
Relation
+ getters and setters()
+ constructor()
1
0..n
name
Menu Item
Id: int
Name: String
Amount: Double
+ getters and setters()
+ constructor()
+ list modifiers()
Balance
+ b_id: int
+ b_expend: double
+ b_taxes: double
+ b_earn: double
+ b_date: date
+ eco_id: int
+ col: int
+ getters and setters()
+ constructor()
Category
+ id: int
+ name: String
+ getters and setters()
+ constructor()
name
Product
+id: int
+name: String
+Item list: list
Category: id
Price: double
+getters and setters()
+constructor()
+subtractConstituents()
Worker Interface
+ Buttons and options
+ optionSelected(View Prod
+ optionSelected (View Orde
+ optionSelected(Add Orde
+ optionSelected(Remove C
+ optionSelected (Log Out)
+ optionSelected (Return)
Manager Interface
+buttons and options
+select(View Users)
+select(Add New Worker)
+select(View Reports)
+select(View Orders)
+Select(Request Report)
name
Admin Interface



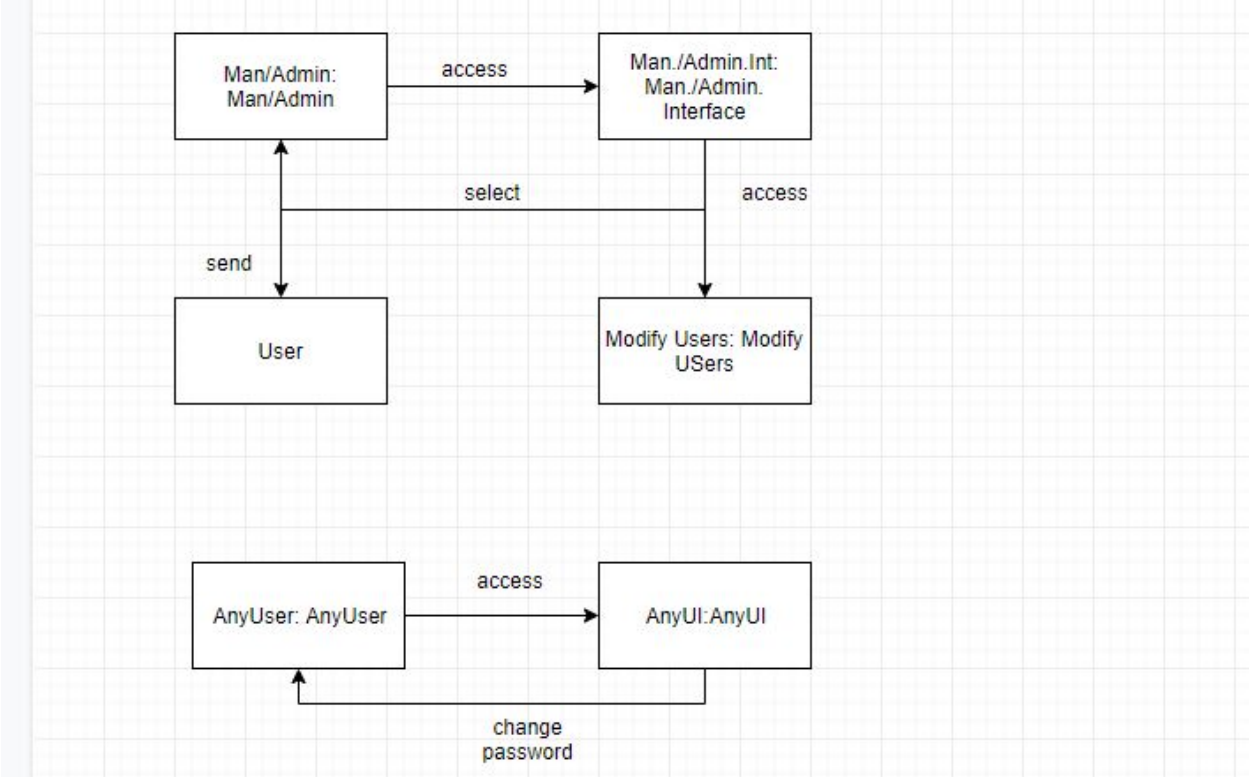
Cappuccino: Collaboration Diagrams


shopWorker: Shop Worker
access
shopWorkerInt: Shop Worker Interface
putOrder
shopWorker: Shop Worker
access
shopWorkerInt: Shop Worker Interface
removeOrder
shopManager: Shop Manager
access
shopManInt: Shop Manager Interface
makeReport
report: Report

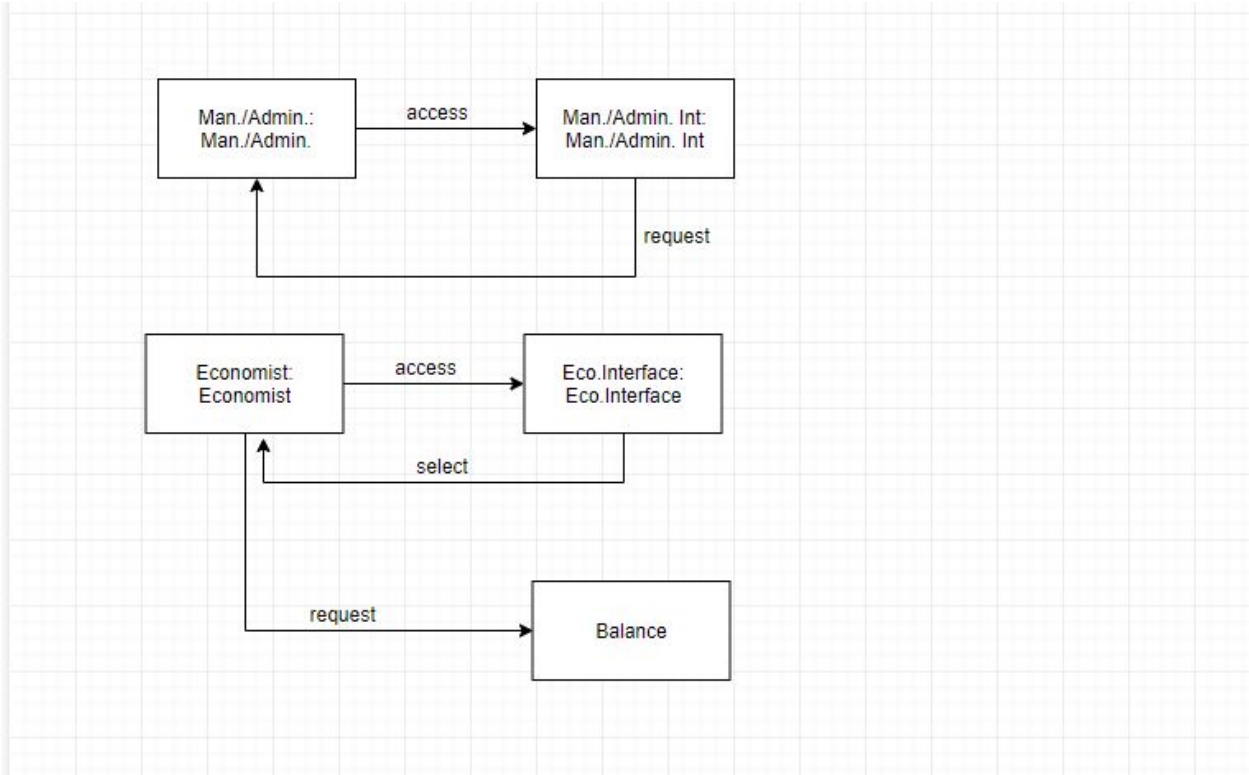
COLLAB 01-03

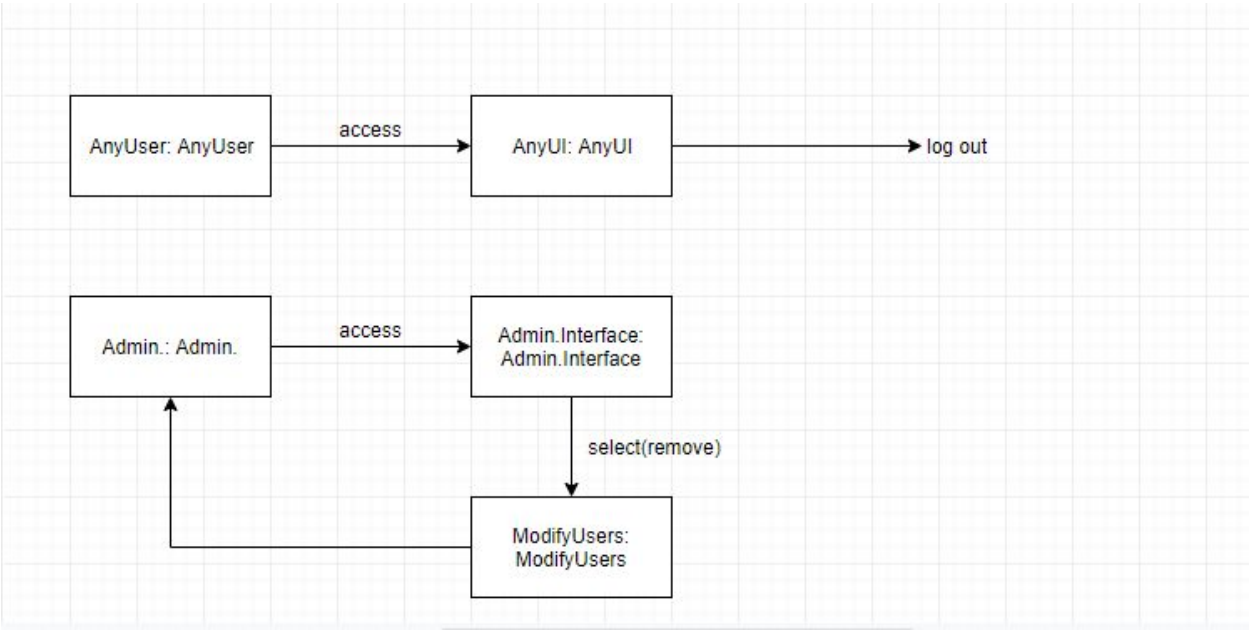

Admin/Manager: Admin/Manager
access
Admin/Manager UI: Admin/Manager UI
modify
createUser
ModifyUsers: ModifyUsers
NewUser: User

SEQ 04

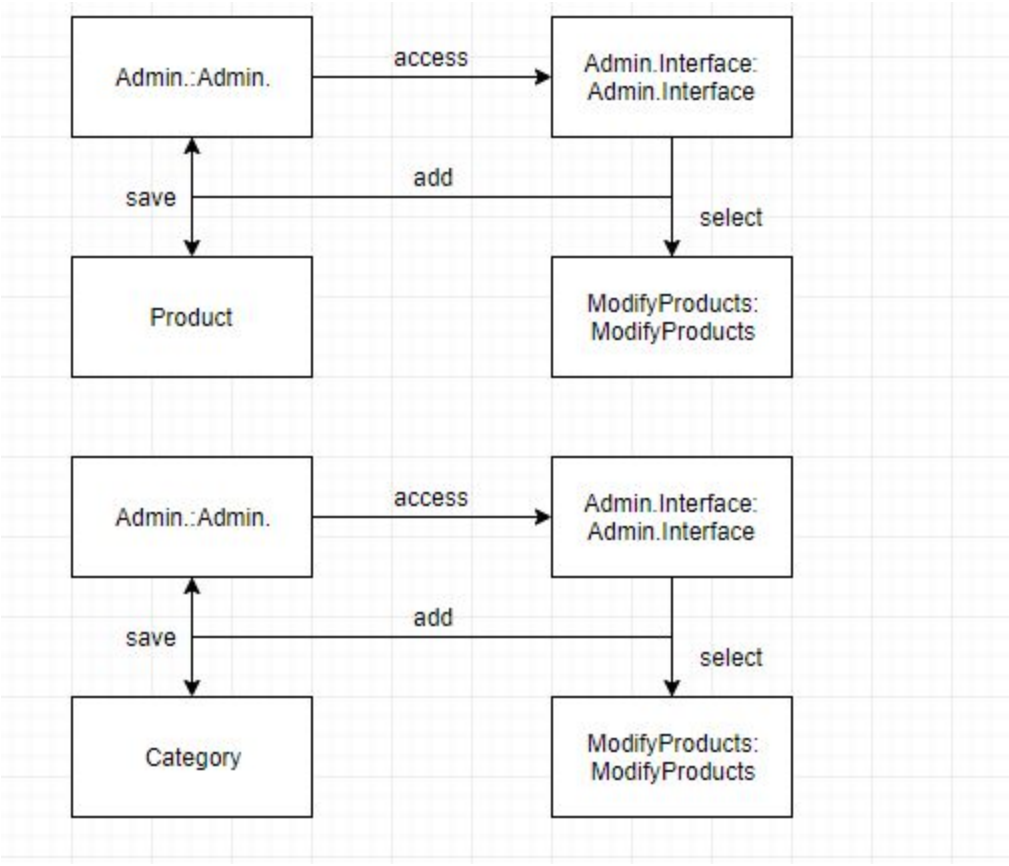


SEQ 05-06

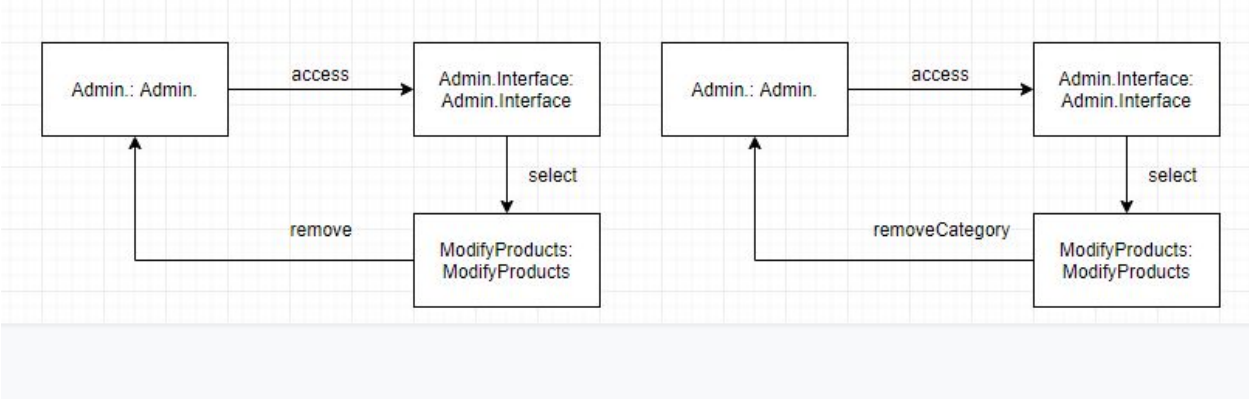




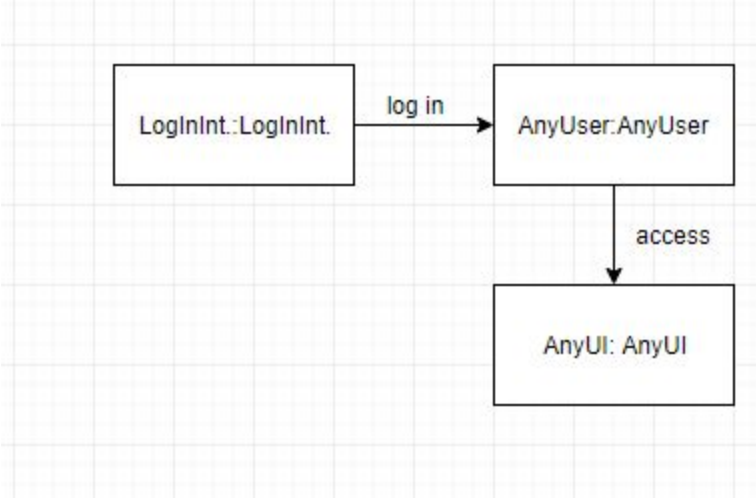
SEQ 07-09



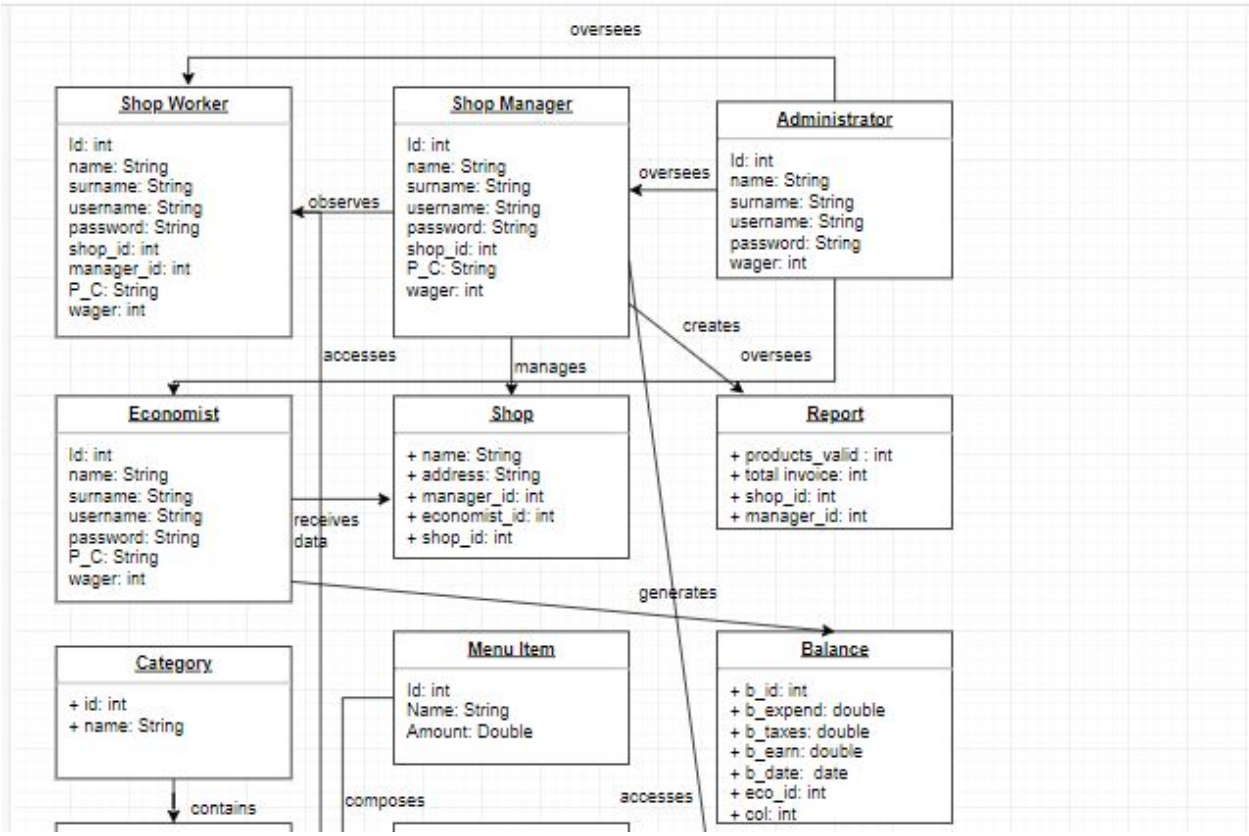
SEQ 10-11

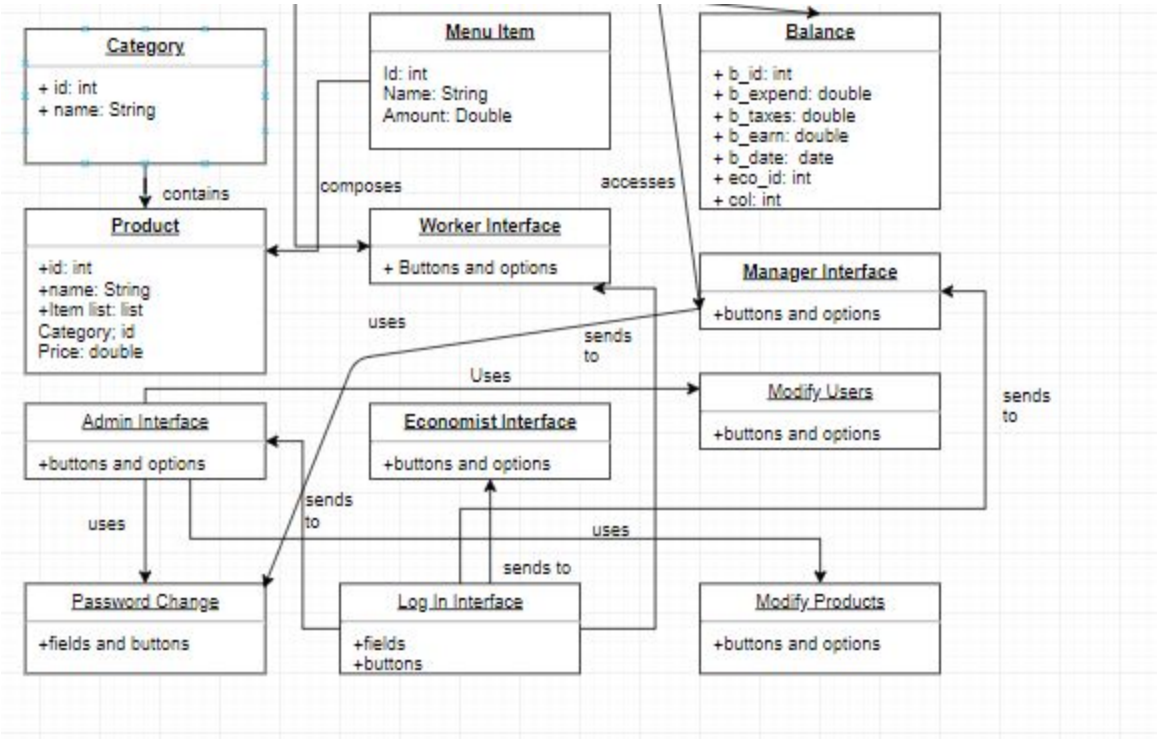


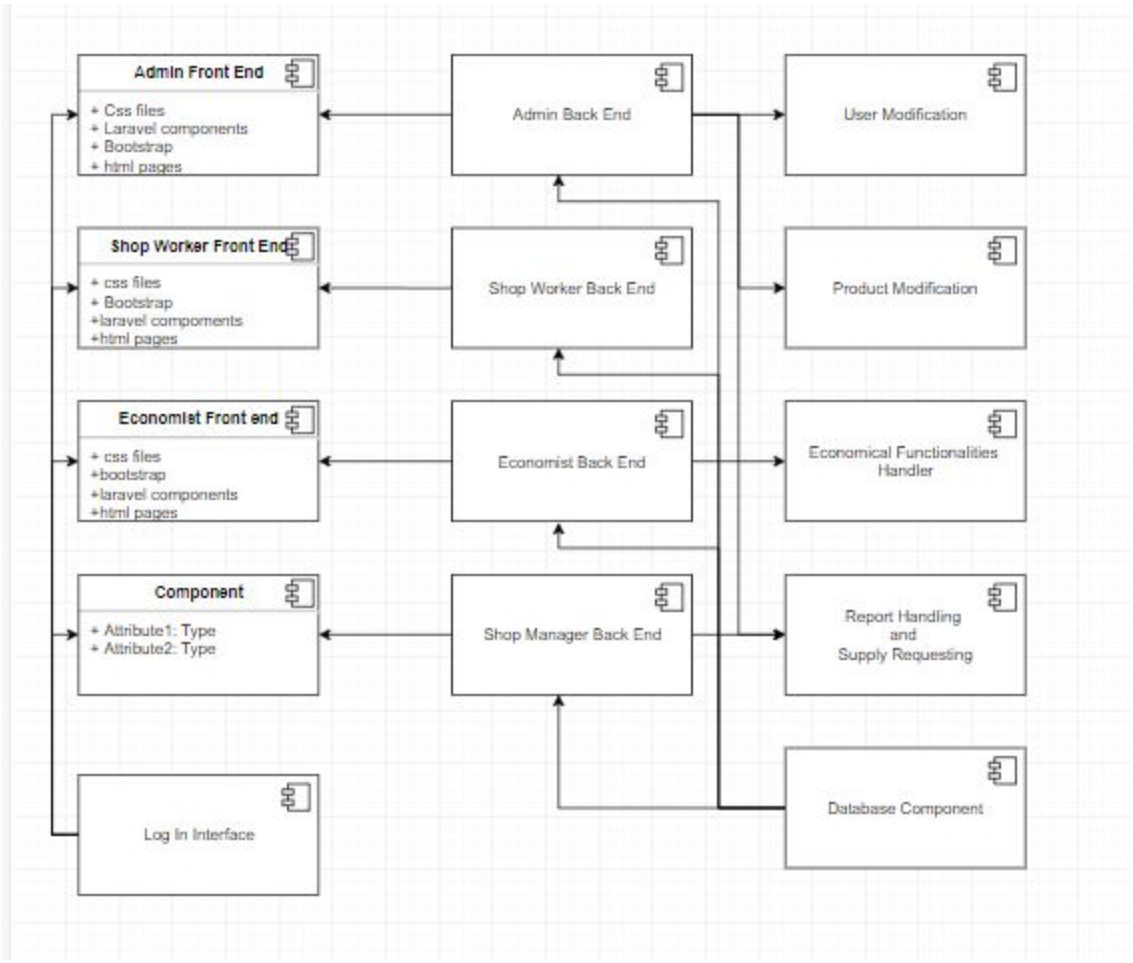
SEQ 12,13

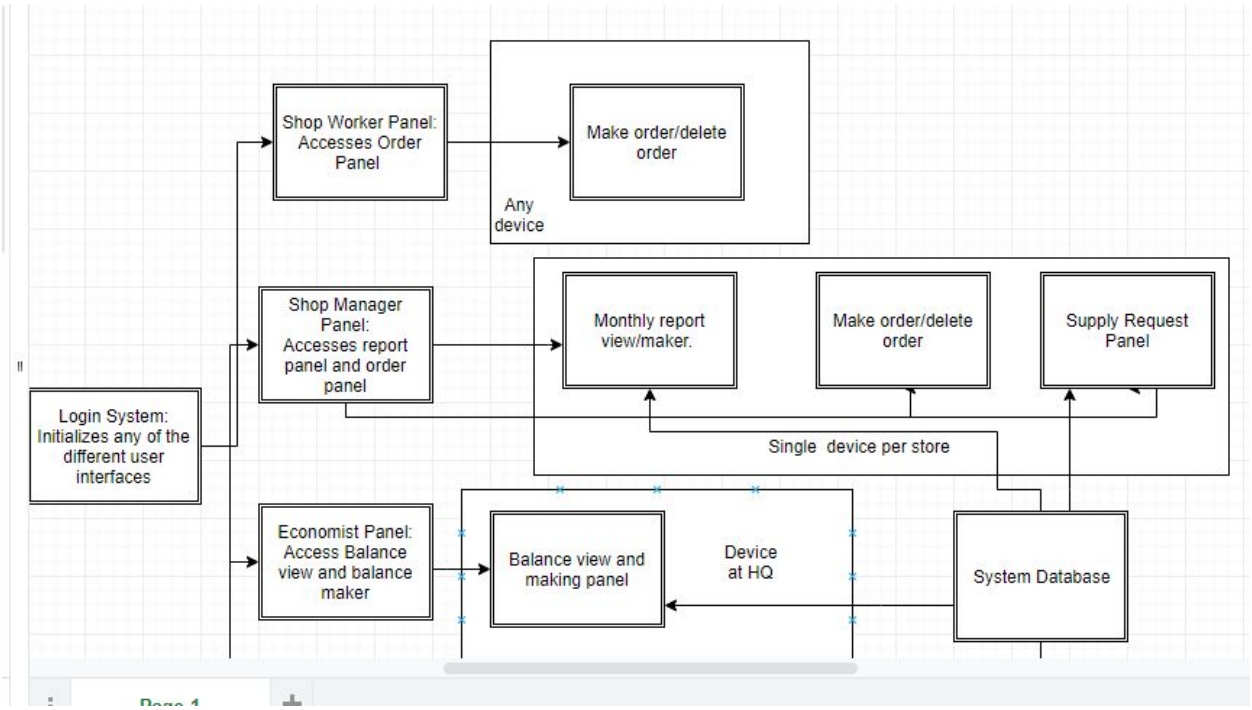


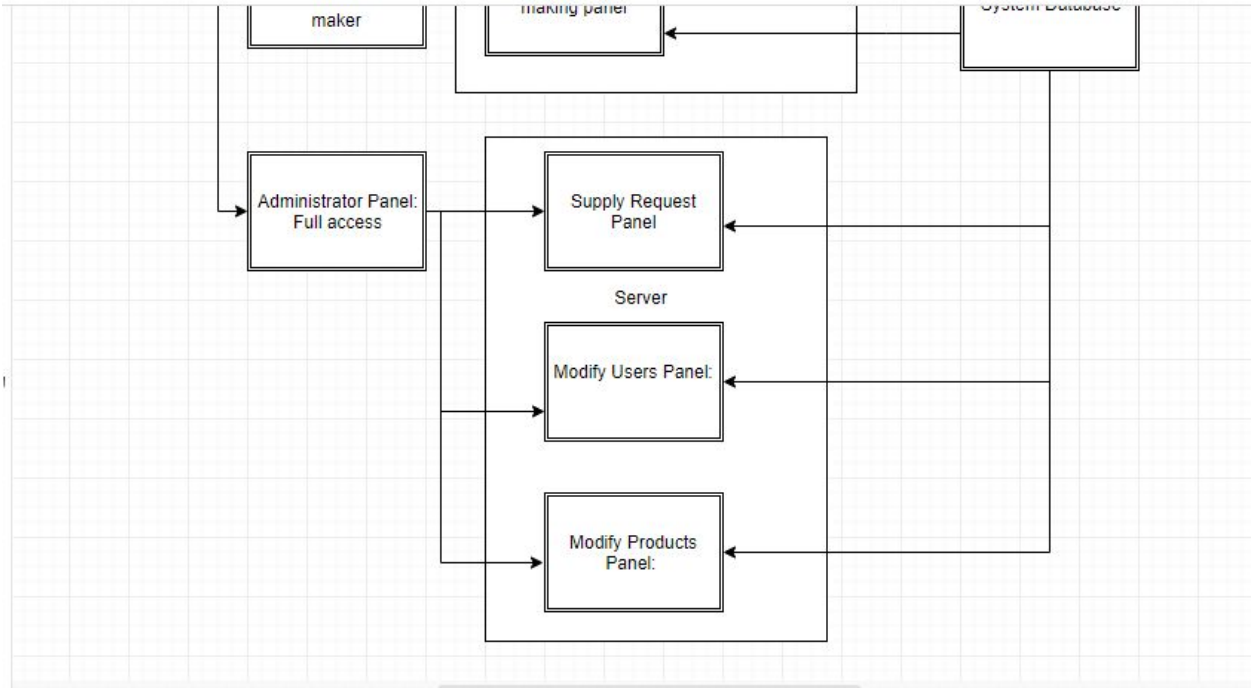
SEQ 14

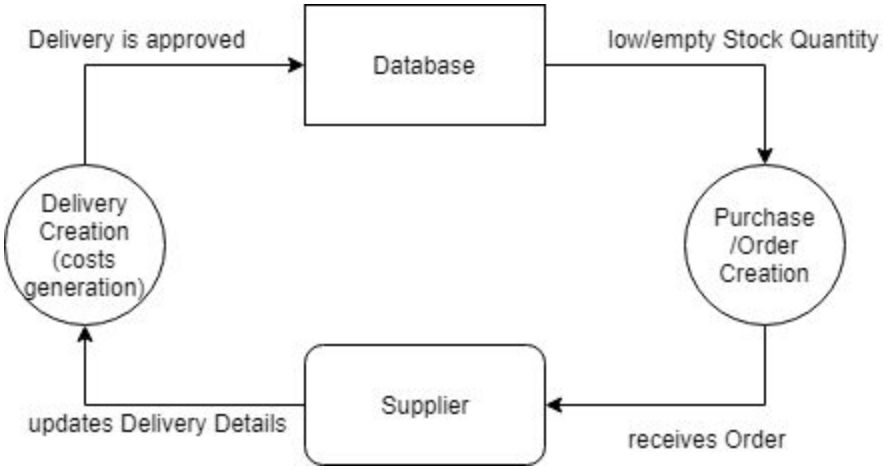

oversees
Shop Worker
Id: int
name: String
surname: String
username: String
password: String
shop_id: int
manager_id: int
P_C: String
wager: int
Shop Manager
Id: int
name: String
surname: String
username: String
password: String
shop_id: int
P_C: String
wager: int
Administrator
Id: int
name: String
surname: String
username: String
password: String
wager: int
observes
oversees
creates
accesses
manages
oversees
Economist
Id: int
name: String
surname: String
username: String
password: String
P_C: String
wager: int
Shop
+ name: String
+ address: String
+ manager_id: int
+ economist_id: int
+ shop_id: int
Report
+ products_valid : int
+ total invoice: int
+ shop_id: int
+ manager_id: int
receives
data
generates
Category
+ id: int
+ name: String
Menu Item
Id: int
Name: String
Amount: Double
Balance
+ b_id: int
+ b_expend: double
+ b_taxes: double
+ b_earn: double
+ b_date: date
+ eco_id: int
+ col: int
contains
composes
accesses

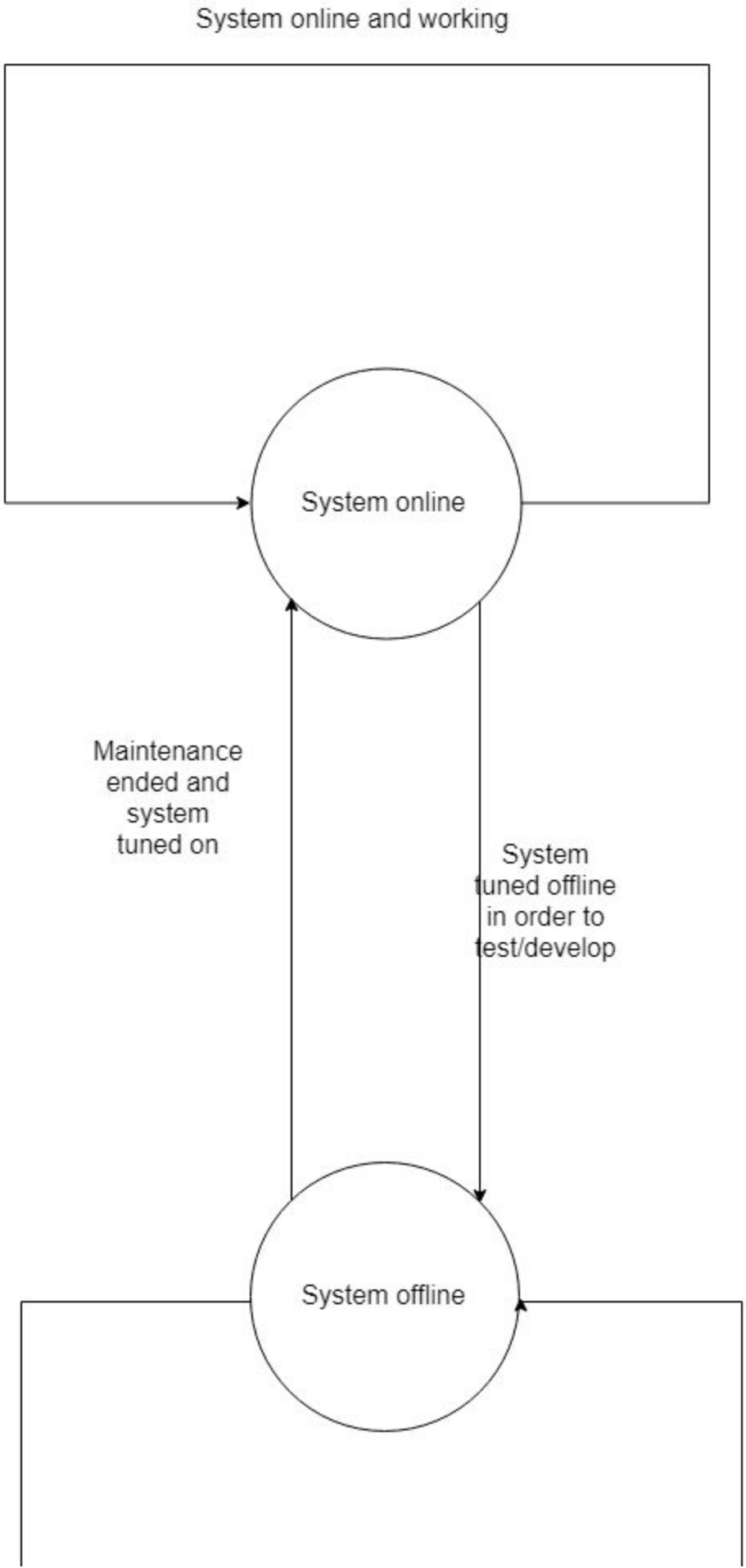

Category
+ id: int
+ name: String
contains
Menu Item
Id: int
Name: String
Amount: Double
composes
Balance
+ b_id: int
+ b_expend: double
+ b_taxes: double
+ b_earn: double
+ b_date: date
+ eco_id: int
+ col: int
accesses
Product
+id: int
+name: String
+Item list: list
Category: id
Price: double
Worker Interface
+ Buttons and options
Manager Interface
+buttons and options
uses
sends
to
Uses
Modify Users
+buttons and options
sends
to
Admin Interface
+buttons and options
Economist Interface
+buttons and options
sends
to
uses
uses
sends to
Password Change
+fields and buttons
Log In Interface
+fields
+buttons
Modify Products
+buttons and options

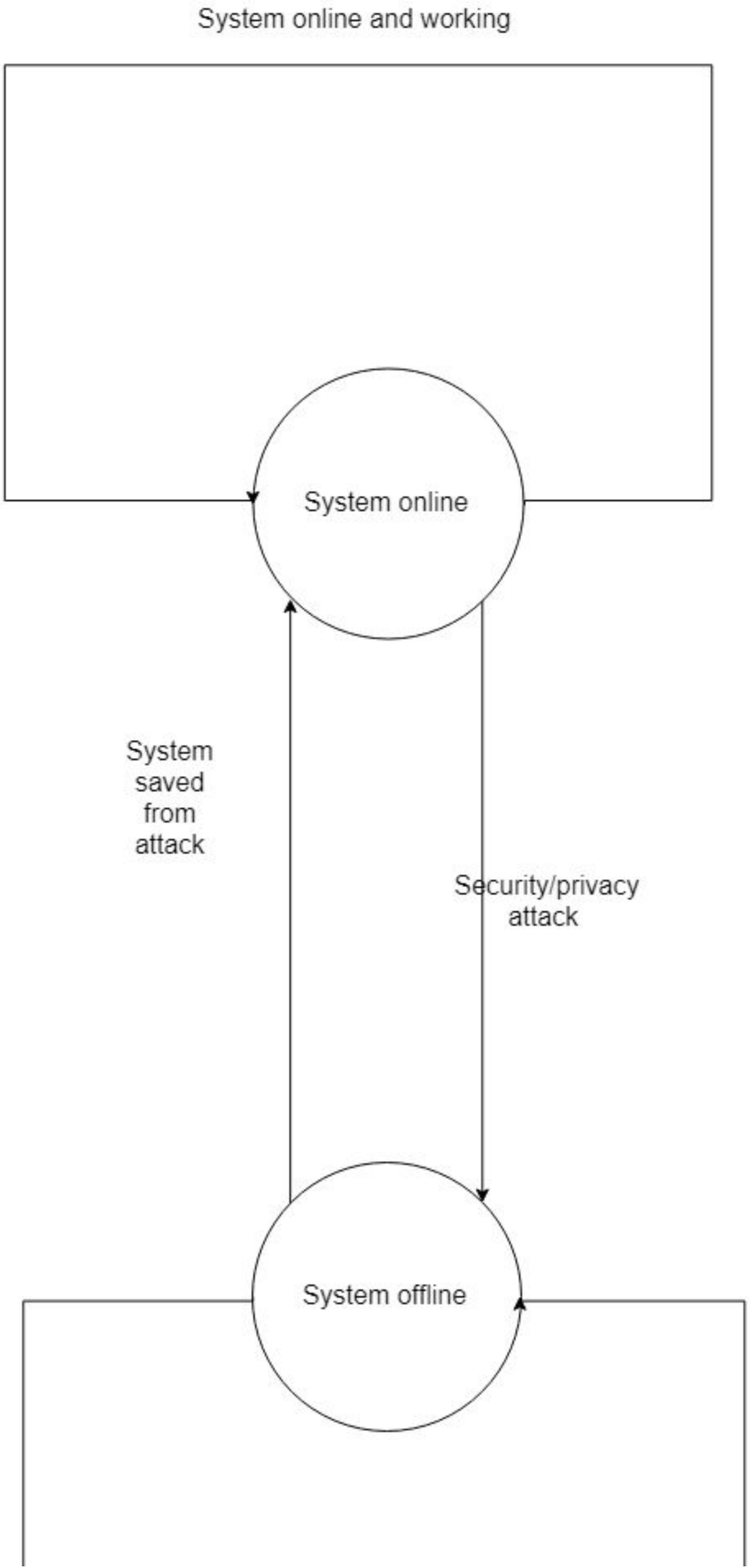

Admin Front End
+ Css files
+ Laravel components
+ Bootstrap
+ html pages
Admin Back End
User Modification
Shop Worker Front End
+ css files
+ Bootstrap
+laravel components
+html pages
Shop Worker Back End
Product Modification
Economist Front end
+ css files
+bootstrap
+laravel components
+html pages
Economist Back End
Economical Functionalities Handler
Component
+ Attribute1: Type
+ Attribute2: Type
Shop Manager Back End
Report Handling and Supply Requesting
Log In Interface
Database Component

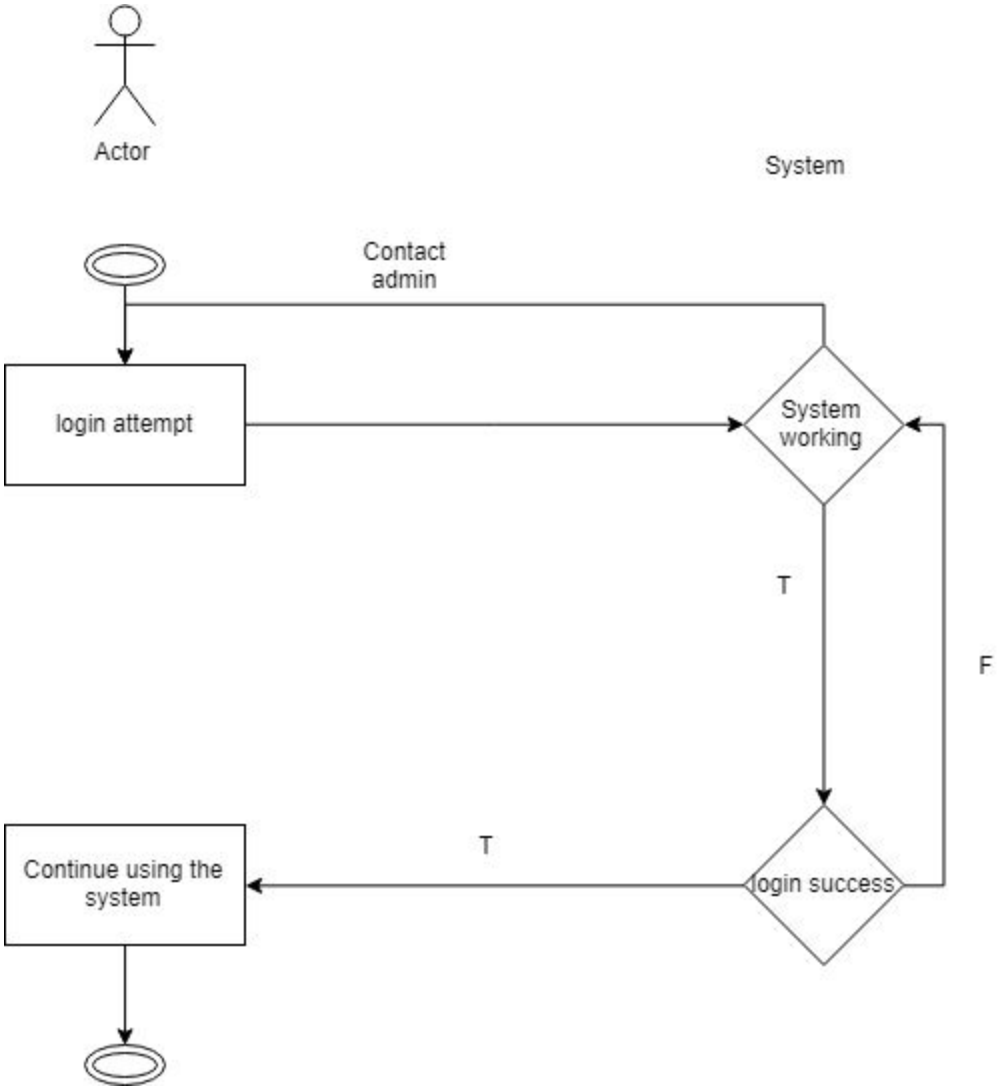

Shop Worker Panel: Accesses Order Panel
Make order/delete order
Any device
Shop Manager Panel: Accesses report panel and order panel
Monthly report view/maker.
Make order/delete order
Supply Request Panel
Login System: Initializes any of the different user interfaces
Single device per store
Economist Panel: Access Balance view and balance maker
Balance view and making panel
Device at HQ
System Database

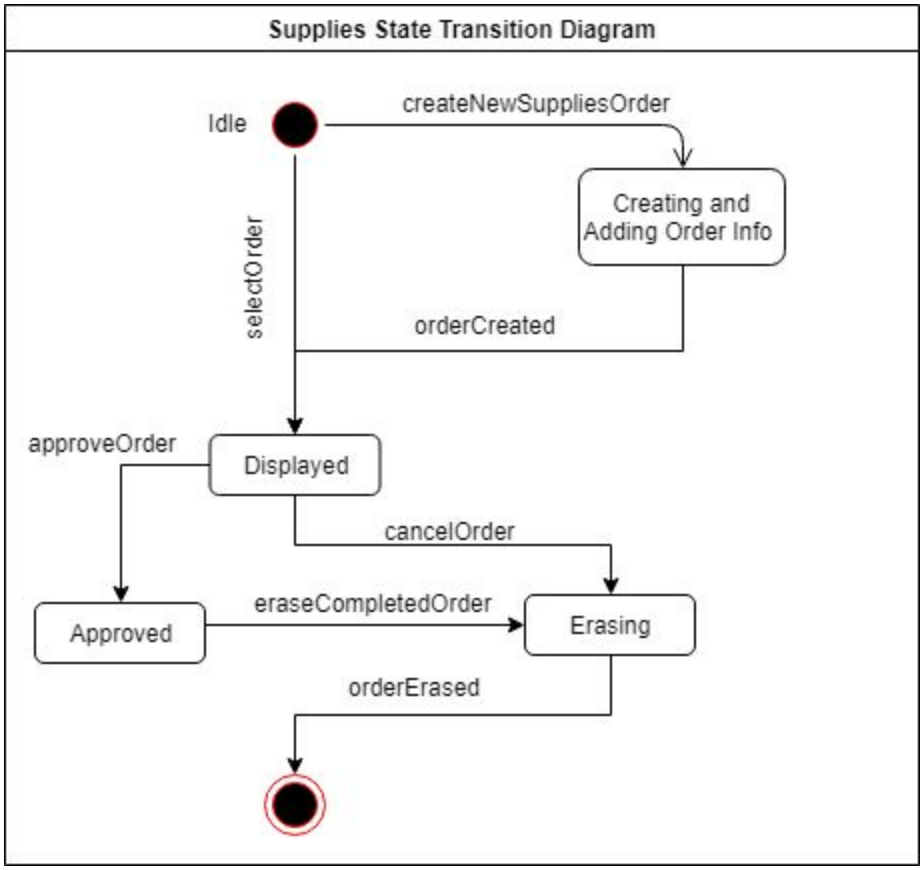

maker
Administrator Panel: Full access
Supply Request Panel
Server
Modify Users Panel:
Modify Products Panel:


Delivery is approved
Database
low/empty Stock Quantity
Delivery Creation (costs generation)
Purchase /Order Creation
Supplier
updates Delivery Details
receives Order


System online and working
System online
Maintenance ended and system tuned on
System tuned offline in order to test/develop
System offline


System online and working
System online
System saved from attack
Security/privacy attack
System offline


Actor
System
Contact admin
login attempt
System working
T
F
Continue using the system
T
login success


Supplies State Transition Diagram
Idle
createNewSuppliesOrder
Creating and Adding Order Info
selectOrder
orderCreated
approveOrder
Displayed
cancelOrder
Approved
eraseCompletedOrder
Erasing
orderErased