Alpha Consult:

Arena FinTech Information System Consult Project

Design Phase Report

Client: Jay Yu , CFO, Arena FinTech, Inc.

Authored By:

Lijun Yang

Nai-Jieh Wang

Ruoyi Li

Yiwen Wei

Yuheng Zhong

05/08/2019

Table of Content

1. Executive Summary

Alpha consult recommended that the client, Arena FinTech to administrate Alpha's proposed back-end administration system along with all updates and improvements. Under the guidance of the agile methodology, Alpha consult team had successfully implemented planning, analysis and design phases through which the team had thoroughly inspected and surveyed the current system of Arena FinTech to draw organizational objectives, problems, opportunities, and directives. Simultaneously, the team fully understood the drawbacks of lacking a platform back-end administration system for platform manager to manage user account information, transaction records, and monitor platform status.

The agile methodology was followed not only to meet the expectations of client and users, but also to ensure that the project will add value to the Arena FinTech at enterprise level in the long run. This new platform back-end administration system will prove to be better and faster than the current account management process. Currently, platform managers need to issue service tickets every time they want to view or edit the user account and transaction record data, which is very inefficient. Alpha consult team advocated for the exploiting the opportunity to rebuild the platform management process by technology.

Through planning, analysis, and design phases of agile methodology, Alpha consult team carefully analyzed the problems and opportunities. By meeting with Jay Yu (CFO of Arena FinTech), Alpha consult team collected a set of business requirements outlining the functionalities of the new system. Detailed technical specifications can be generated by studying the business requirements list. In light of the overall scope of this project, the business and technical requirements inherently present, as well as the current problems and opportunities, we believe that the team can help Arena FinTech to move a step forward with the development of this project.

2. Evaluation letter

May 1, 2019
Mr.Jay Yu
Chief Financial Officer
Arena FinTech, INC.

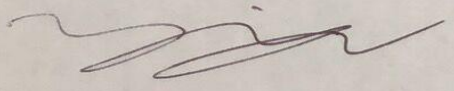
Dear Mr. Berecz,

This letter of evaluation regards the whole team participated in the Arena FinTech information system consult project; the members of the group are Yuheng Zhong, Lijun Yang, Nai-Jieh Wang, Ruoyi Li and Yiwen Wei.

Yuheng and his team were responsible for building the backstage administration system and the interface. They actively discussed every detail of the system with us, confirmed our requirements and needs and recorded our feedback. The project worked quite well with their efforts.

On behalf of Arena FinTech's management, I would like to extend our appreciation of the amazing work done by Yuheng and his team. The endless time that they spent on this project, and the professionalism that they portrayed have impressed us and we deem ourselves worthy to have them as in the project.

Best Regards,



3, Approved SOW

3.1 Client and Industry Background

Arena FinTech, INC.

Arena FinTech, INC. focuses on providing a well-designed multifunctional forex information and trading services platform to forex traders. Their services include forex trading, finance analysis, stock trading analysis, and forex information analysis. Arena FinTech, INC. was founded by Scotte Huang in Los Angeles, California in 2018.

Arena FinTech Business process

Arena FinTech acts as the middleman between forex traders and foreign exchange companies. Once forex traders create a transaction on Arena forex brokerage platform, Arena will send a trading request to Oanda (Foreign exchange company) to process the transaction. Oanda will send back a trading result to Arena and Arena will charge 10% service fee from their users if the trading result is "Success". In order to attract more and more people to Arena FinTech platform to register and conduct a forex transaction, Arena FinTech developed various forex trading features such as Earning rank list, subscribe trading, and investment analysis for their users to use.

Forex brokerage and information service industry

According to resources posted by James Chen[1], forex brokerage refers to those firms who are providing a forex trading platform between forex traders and foreign exchange companies. The users can access to the currency market to sell and buy foreign currencies on the platform. The forex broker will charge a service fee per transaction to gain profit. The entire forex brokerage market is super huge with average daily volume hitting $18.9 billion in 2018 and this number is still going up[2] .

[1] https://www.investopedia.com/terms/forex/c/currency-trading-forex-brokers.asp
[2] http://fairreporters.net/economy/largest-forex-brokers-by-volume-in-2015/

3.2 Current system problems/opportunities

Problem: low management efficiency

Arena FinTech does not have a backstage administration system, all management including database management and account management are done by IT staff using terminal code. For instance, if the manager want to change the authorization privilege, the manager need sending a request ticket to IT support. IT support people will modify the code to modify the authorization privilege once they receive the ticket. The whole process is time-consuming (usually take 2~6 hours) and very user-unfriendly to non-technical people.

Opportunity: improve customer experience

1.Multi-language interface for the platform

Forex is a global decentralized market with highly-diversified customers. In order to improve customer experience, creating a multi-language platform is a good opportunity to meet needs of international customers.

2.Simplify transaction process

Customers have to go through cumbersome processes to acquire authorization for each transaction. To be specific, they have to go to the firm's website, like OANDA, to verify their accounts to the platform by themselves when making each transaction, instead of acquiring authorization directly from the platform.

3.3 Project Scope

Initial Scope

The project scope is determined by our current understanding of Arena's needs. Our client raised three aspects for us to improve. First, a backstage administration system for managers which includes a database of client information and a web-based application. Second, multi-language interface for the platform which can support languages apart from English according to users' preferences. Third, a pop-up login function which allow users to login directly on the platform and not be transformed to another website.

Final Scope (After correction)

Due to time limitation of the project timeline from February to May and the complexity of creating an API for system, we will only focus on building the backstage administration system and its interface. Pop-up login and multi-language functions are dropped out of our scope.

The technology requirements of the system include the usage of web development languages, for example, Django, HTML, CSS, etc. For database, we will use MySQL.

3.4 Project Objectives

The objectives of the project are to publish backstage administration system to satisfy the client's requirements. New products will help improve clients' working efficiency and terminate management problems.

There are several elements of what constitutes all user stories being "success", as in a completed feature. We employ these criteria to measure the success of the proposed solutions.

These critical success factors Included:

1.Detailed and feasible update and maintenance plans of interface are made.

2.The backstage administration system, the new database and the application, are developed successfully.

3.The backstage administration system can be used by all managers to handle daily mass data.

4.The backstage administration system is consistent with updates of front-stage interface.

5.The backstage administration system shortens the original 2~6 hours business process to instantaneous completion.

6.Both the interface and the system maintain functional integrity while ensuring that time and budget constraints.

7.Both the interface and the system have passed User Acceptance testing and are signed off by the client.

8.All deployment documentations are produced or updated.

9.Remaining story points for the projects are set to zero and the project is closed.

3.5 Project Constraints

The Constraints of the project lie in two aspects

1. Confidentiality

Our team plans to build a new database for the backstage administration system, so classified information is required from our client, including customers' login information. Client may have concerns about our access to these information and not willing to provide them. If the resources are not available, time to deliver may increase. Also, team members have to pay extra effort and time to ensure privacy.

2.Time limitation

Due to the complexity of the project, time becomes a problem and may restrict our performance. We will try our best to meet client requirements and actively communicate with client to adjust our solutions.

3. Technical concerns

Since we have to build a complete new backstage administration system and the corresponding user interface, we need to learn the development languages and its application in a short time. We are going to meet problems on applying certain functions that we want, and this may slower our progress.

3.6 Client Approval

Sign: 

Date: 02/23/2019

By Jay Yu – CFO, Arena FinTech, INC.

Sign: 

Date: 02/23/2019

By Yuheng Zhong – Team Lead, Alpha Consult.

4. Analysis Phase Report

4.1 Fact finding and requirement gathering

According to the location of our client (California) and the nature of Arena FinTech project, our team decided to use interview via Skype as the fact-finding method and user story as the requirement gathering method.

Interview

Before the interview, our team performed a comprehensive analysis on current system's issues and proposed solutions in SOW. In order to perform a high efficiency interview with our client, we prepared a list of interview questions and meeting agenda for our client.

The interview took place on March 20th via Skype. We started the interview with a short discussion on the issues in current system. Ruoyi, Lijun, and Shirley took notes on them and generated a draft data flow diagram of current system. Naijieh and Yuheng collected the Database information and generated a draft ERD of current system.

After defining the current system process and restating the proposed solution, we begun to ask our client—Jay, the questions from the interview list to gather client requirements. Ruoyi took notes for answers of interview questions.

After the interview, our team reviewed the meeting notes and interview answers to generate several user stories and stored them as the product backlog on the Trello board. Client can review and make comments on these user stories anytime from Trello board.

User Story

According to the definition posted by Margarent Rouse[3] , user story is a Top-Down requirement gathering method to capture a description of a system feature from an end-user perspective. User story starts with an epic, which means how the entire system looks like, and breakdown into several detailed end-user descriptions. User story uses the following format:

- As a <role>, I want <feature> so that <reason>

Trello board product backlog

[3] https://searchsoftwarequality.techtarget.com/definition/user-story

Trello board is the main tool for us to manage the project and collaborate with our client. After the interview, all collected user stories have been stored as the product backlog on trello board. Jay can review the project progress and make comments anytime on the trello board.

Interview Questions

List of questions

- What do you like about the layout?
- What do you like about the site map?
- What functions do you want for the administration site?
- Detail of each function
- What do you like about the authorization privilege ?
- Definition of each database
- How many levels of subpage you want for each main page?
- Do you imagine any design elements would hurting the site?

Meeting Agenda

Arena FinTech – Analysis Phase Meeting Agenda

Meeting Information

Client: Jay Yu

Date: 03/20/2019

Location: Skype Meeting

Presenters: Lijun Yang, Naijieh Wang, Ruoyi Li, Yiwen Wei, Yuheng Zhong

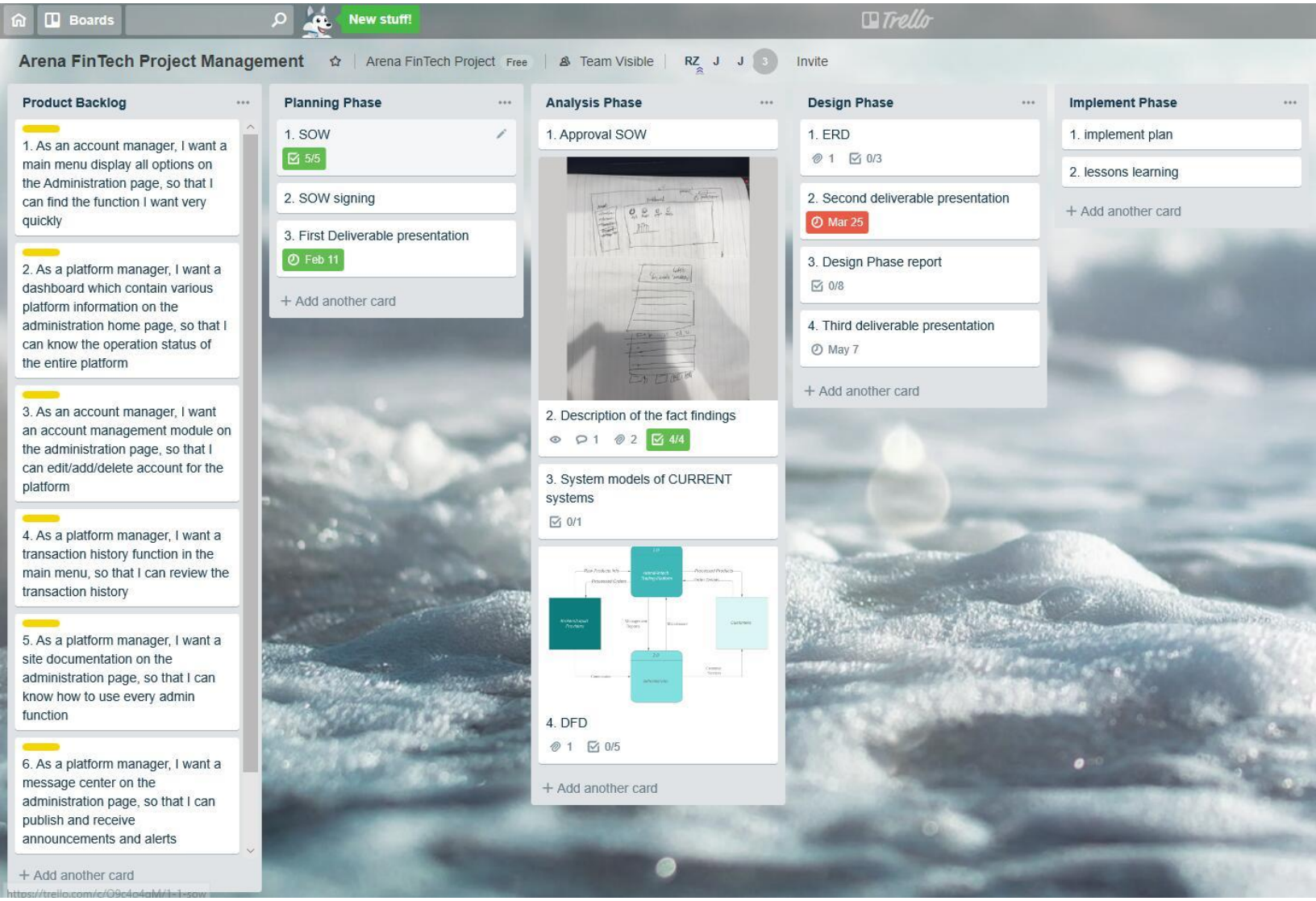
Meeting Documents

- SOW
- Requirement Interview questions
- Product oriental questions & requirement sheet
- Product outline
- Meeting Agenda

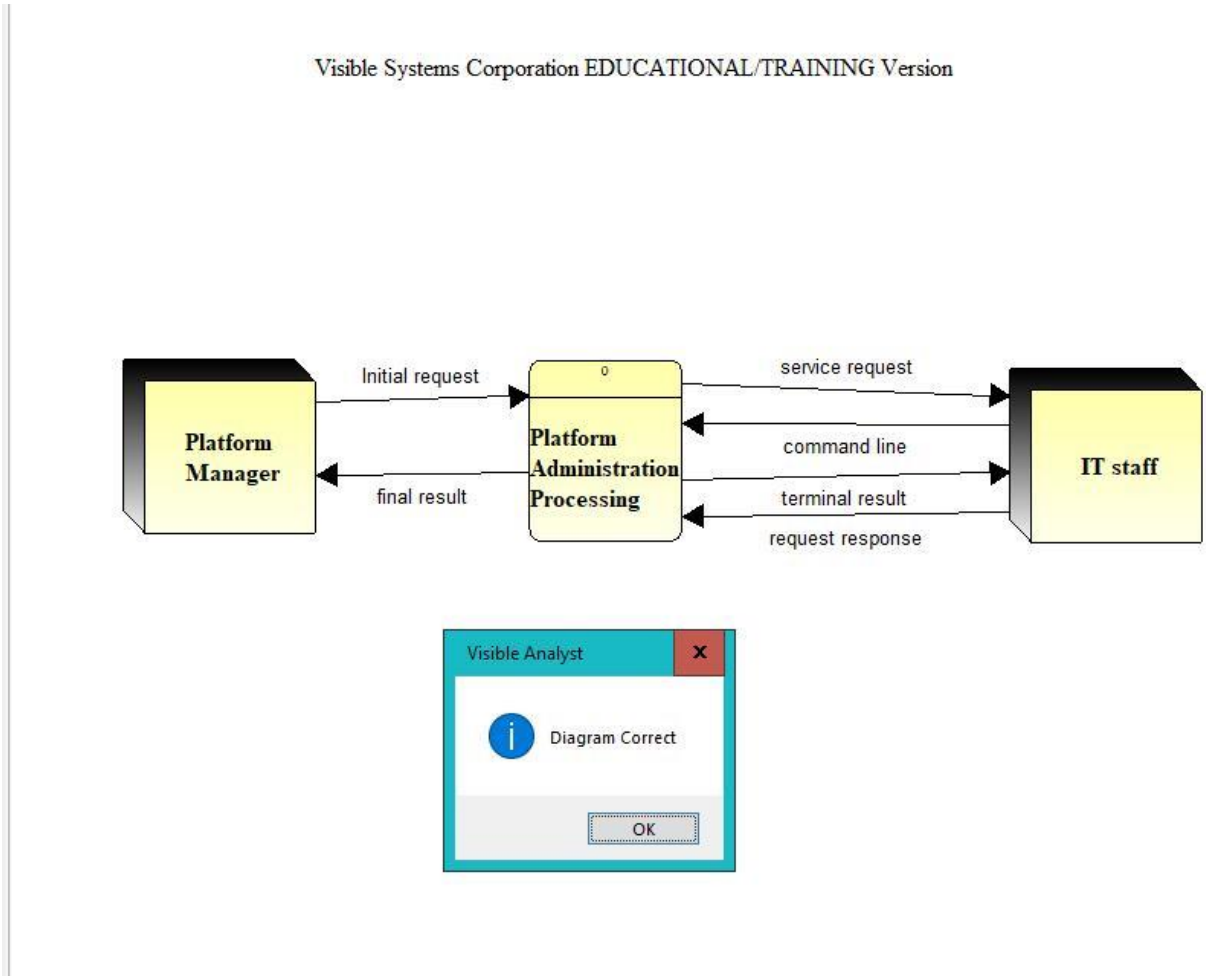
Meeting Process

Step1: Send out meeting agenda & self-introduction	By Yuheng (2 mins)
Step2: Review and correct SOW	By Lijun (3 mins)
Step3: Ask client's goal & requirements	By Naijieh (10 mins)
Step4: Present the outline of product	By Yiwen (10 mins)
Step5: Additional questions & requirements	By Ruoyi (5 mins)

Trello Board Overview



4.2.1 DFD Context level (with VA analyze report)—Current System



4.2.2 DFD Context level-Narrative

The preceding data flow diagram describes the highest level of the current Arena platform administration process between the platform manager and IT staff.

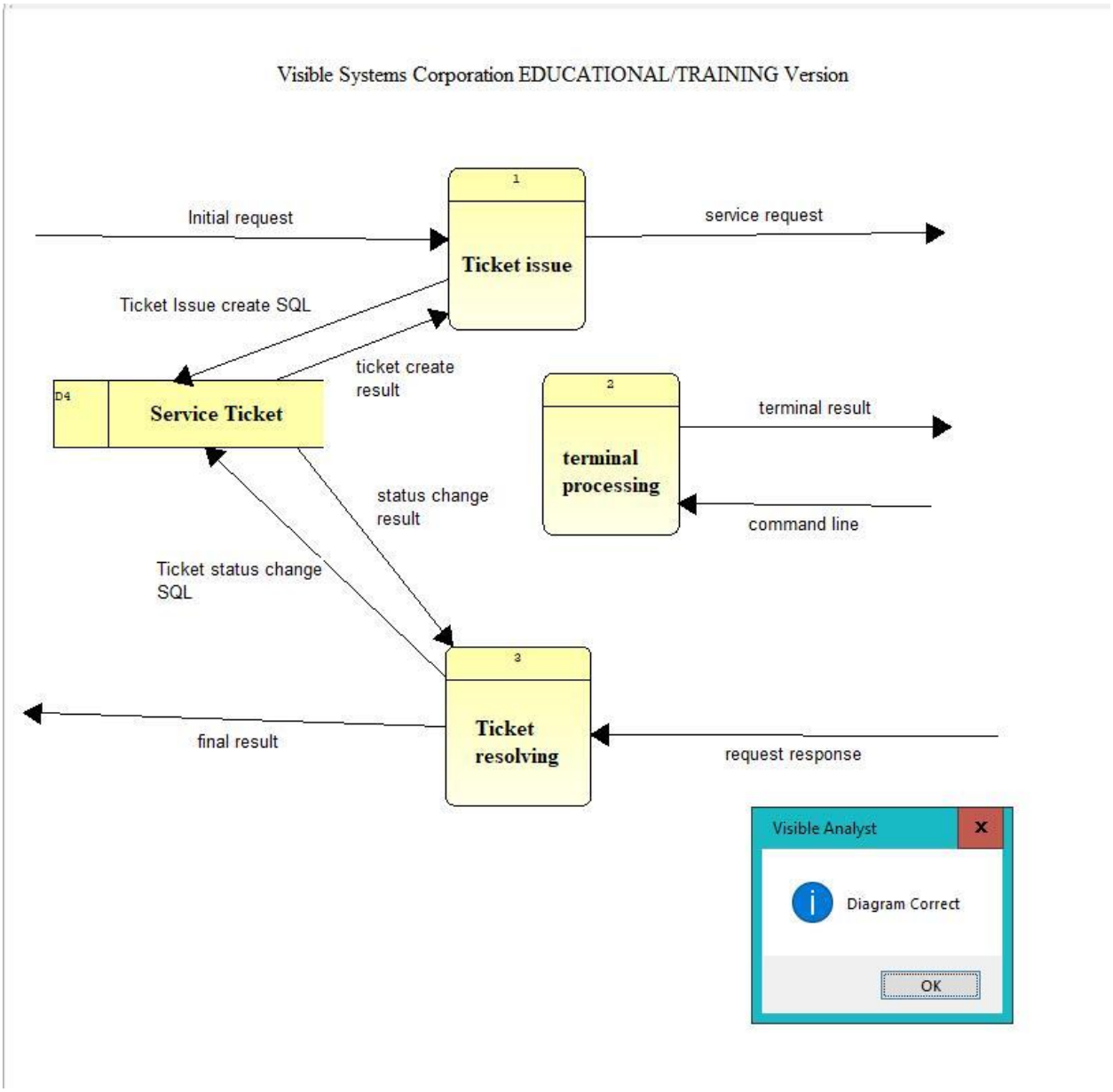
If the platform managers want to modify any platform user account information or review any transaction record, they need to:

1. Initial and fill in a service request ticket, which states the service purpose and detail information.
2. Once the service request ticket has been created, it will be processed and sent to IT staff
3. IT staff will type command code to the system terminal to perform the service after they receive the service ticket
4. Once the command code has been executed successfully by the system, the system terminal will return an execution result to the IT staff.
5. Based on the terminal result, the IT staff will change the ticket status.
6. If the ticket status has been changed to "complete", a final result will be sent back to the platform managers.

4.2.3 DFD Context level-Dictionary

Data Flow Name	Description
Initial request	An initial service request ticket generated by the platform manager
Service request	A formatted service ticket has been processed and sent from the system to IT staff
Command line	System terminal command code typed by the IT staff to perform the service
Terminal result	The result of executed command code typed by the IT staff
Request response	The request response sent by IT staff to the system to update the ticket status
Final result	The final result sent by the system to the platform manager

4.2.4 DFD level 0 (with VA analyze report) —Current System



4.2.5 DFD level 0-narrative

The preceding data flow diagram describes the highest level of the current system employed by managers and IT staff revise the records in the system.

1. Initially, managers create their initial service request to start ticket issue process.

   1.1 Ticket issue process will create appropriate SQL query to service ticket database to save request data.

   1.2 Once data is successfully stored in the database, ticket service database will return the result to ticket issue process.

   1.3 The process then proceeds, sending request to IT staff.

2. IT staff interacts with the terminal to fulfill terminal process.

   2.1 IT staff employs command lines to make changes in the terminal.

   2.2 When changed successfully, terminal will notify the IT staff of the result.

3. IT staff responds to managers via ticket resolving process.

   3.1 IT staff returns the request response to notify managers.

   3.2 Ticket resolving process then create ticket status change SQL query to make changes in the service ticket database.

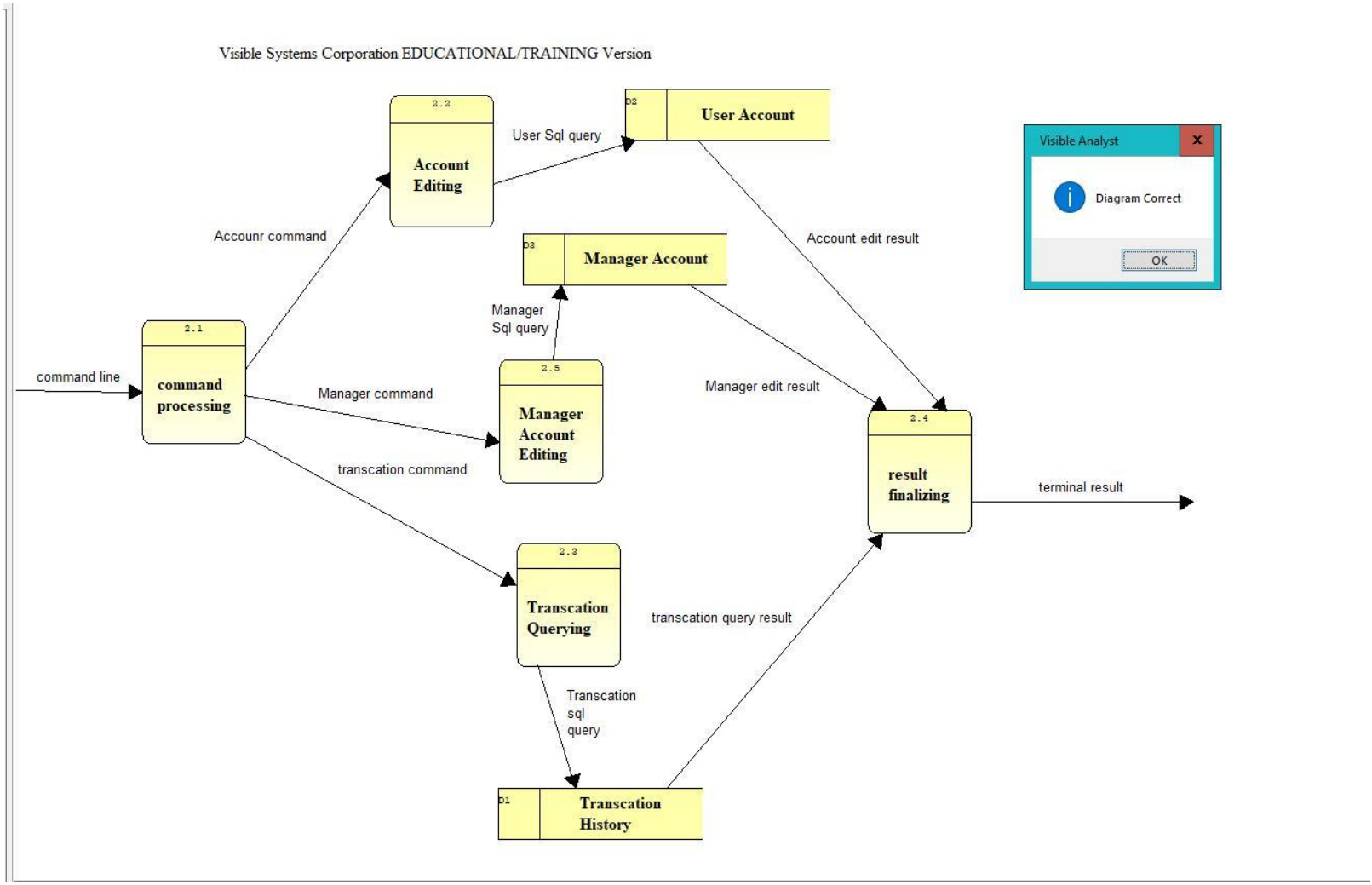
   3.3 When the ticket status has been changed to "complete", database will return the status change result to continue ticket resolving process.

4. Managers finally receive service result from the system.

4.2.6 DFD level 0-Dictionary

Data Flow Name	Description
Ticket issue create SQL	A formatted SQL sentence including ticket information has been created and sent by system
Ticket create results	The result of newly existing data in service ticket database is returned to the system
Ticket status change SQL	A formatted SQL sentence including status change command sent by system to make changes in the database
Status change result	The result of changing status in the database

4.2.7 DFD level 1 (process 2.0) (with VA analyze report) —Current System



4.2.8 DFD level 1 (process 2.0)-narrative

The preceding data flow diagram describes a level 1 breakdown of the process 2.0-terminal process. After the IT staff receive a service request from the system, they will:

1. Typing initial command line to connect to different database /applications based on the purpose of the service

2.1 If the purpose of the service is to modify platform user account information:

I. UPDATE/ADD/DELETE sql query will be sent to the user account database to be executed

II. Once the sql query has been executed successfully, account edit result will be sent back to IT staff

2.2 If the purpose of the service is to modify manager account information:

I. UPDATE/ADD/DELETE sql query will be sent to the manage account database to be executed

II. Once the sql query has been executed successfully, manager edit result will be sent back to IT staff

2.3 If the purpose of the service is to track transaction account information:

I. UPDATE/ADD/DELETE sql query will be sent to the transaction history database to be executed

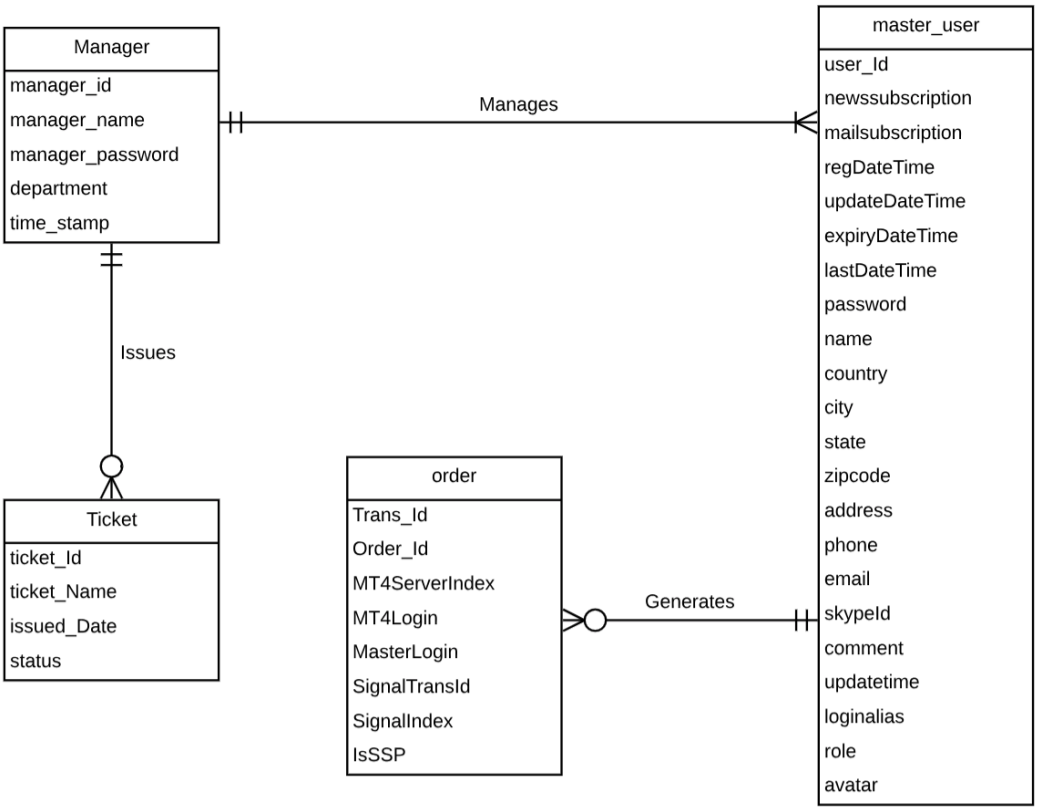
II. Once the sql query has been executed successfully, transaction history result will be sent back to IT staff

3. Finally, after the command process is completed, the terminal result is sent back to the manager for further reviews and comments.

4.2.9 DFD level 1 (process 2.0)-Dictionary

Data Flow Name	Description
Account command	The command code typed by the IT staff to process user account modification
Manager command	The command code typed by the IT staff to process manager account modification
Transaction command	The command code typed by the IT staff to process transaction record exporting
User SQL query	The UPDATE/DELETE/ADD sql query needed to be executed by the database to modifies user account information
Manager SQL query	The UPDATE/DELETE/ADD sql query needed to be executed by the database to modifies manager account information
Transaction SQL query	The UPDATE/DELETE/ADD sql query needed to be executed by the database to modifies transaction history information
Account edit result	The account edit result returned by the system if the command code is executed successfully
Manager edit result	The manager edit result returned by the system if the command code is executed successfully
Transaction query result	The transaction edit result returned by the system if the command code is executed successfully

4.3.1 ERD—Current System



4.3.2 ERD narrative

There are four entities in the ERD diagram. The narratives of each entity are listed below:

Manager:

The entity stores manager account information including ID, name , password etc.

Relationships:

- A manager can manage one or many master_user accounts.
- A manager can issue zero or many tickets.

Ticket:

The entity stores tickets issued by the managers. It contains information about ticket's name, ID and process status.

Relationships:

- A ticket can be issued by only one manager.

master_user:

The entity stores client's account information including user account information and personal contact information.Relationships:

- A master_user is managed by only one manager.
- A master_user can generate zero or many orders.

order:

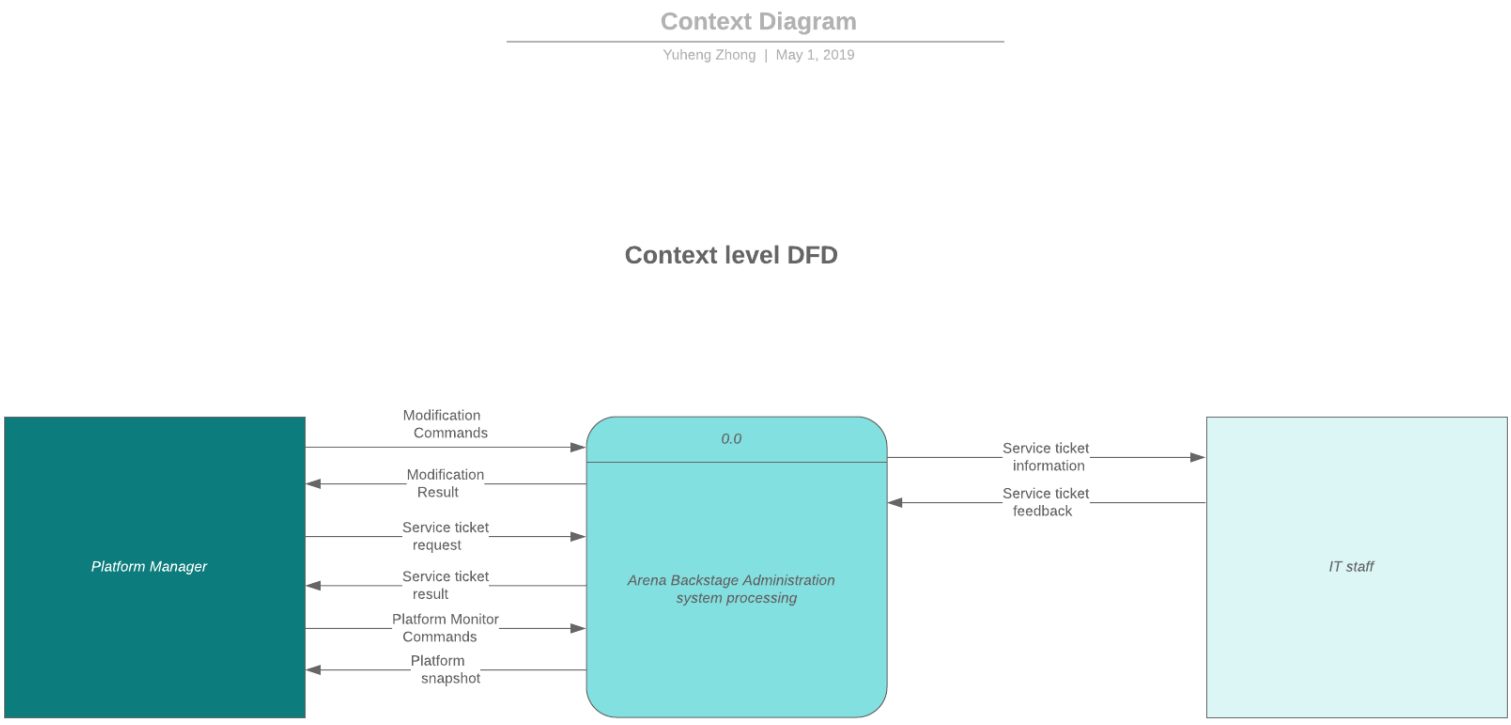
The entity stores order information generated by customer requests. It contains transaction ID, order ID etc.

Relationships:

- An order can be generated by only one user.

5.1 Data Flow Diagram - Proposed System

5.1.1 Context level DFD



Context level Narrative

The preceding data flow diagram describes the highest level of the proposed platform backstage administration system. Same as the current platform management process, there are two external entities involved in the new platform backstage administration system: Platform manager & IT staff. The main difference is that the proposed backstage administration system moves the major of management work from the IT staff side to the platform manager side, which allow platform manager managing the platform directly.

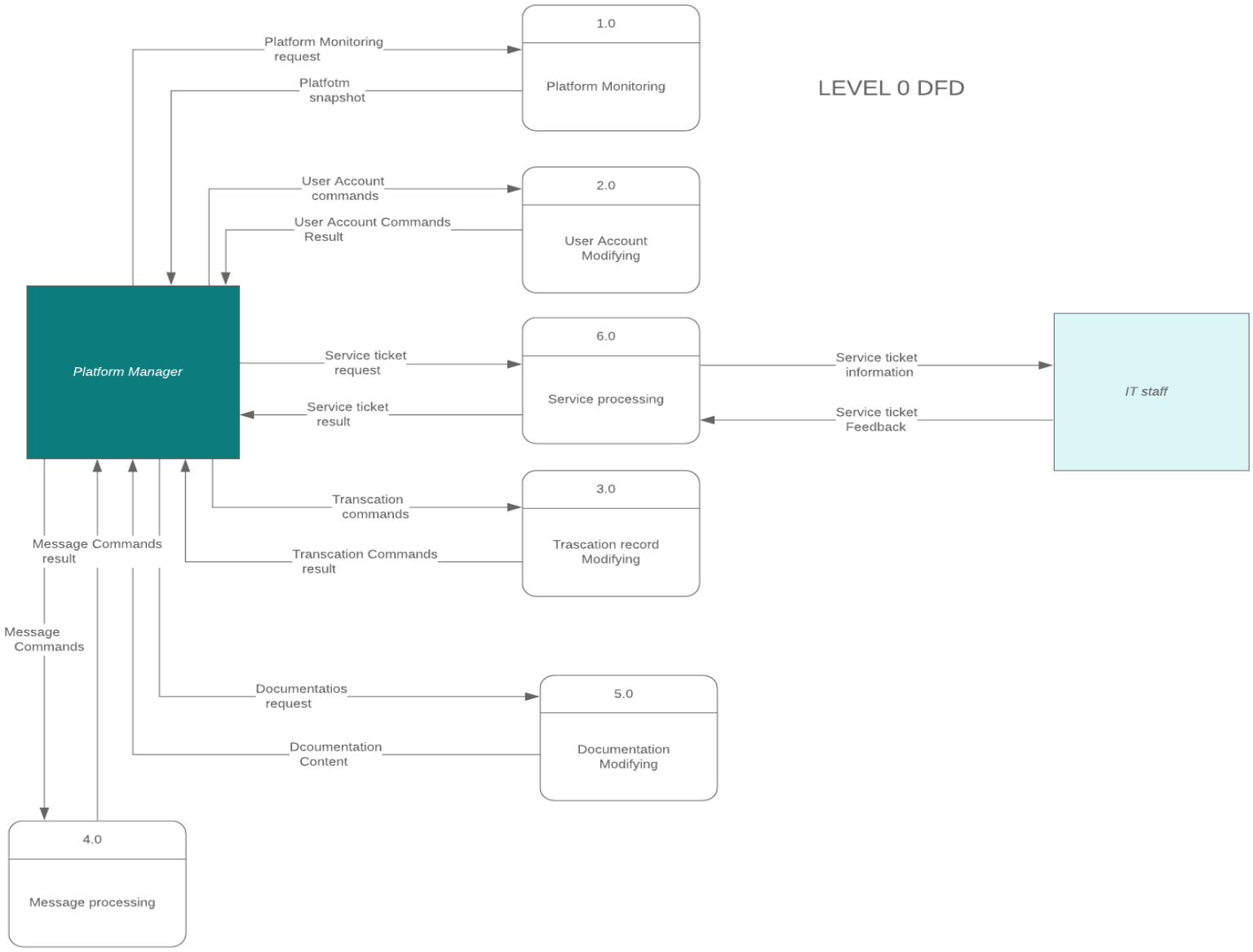
Platform Manager:

1. Send commands to the backstage administration system to view, add, edit, and delete the user account, transaction records, message, and documentation data.
2. Once commands received by the backstage administration system, the data processing will take place in the server side. Processing result will be generated by the system and send back to platform manager.
3. If platform manager finds any platform issues/bugs, manager will initiate a service ticket to the administration system.
4. Once the service ticket status is updated by IT staff, the administration will send a response to platform manager.
5. Platform can send the monitoring command to the administration system to monitor the real time platform situation.
6. Once the monitoring commands received by the administration system, a system snapshot will be generated and send back to manager.

IT staff:

1. Once a service ticket is initiated by platform manager, the administration system will send the ticket information to IT staff to process.
2. Once IT staff complete the service ticket, IT staff will send a ticket update back to the system.

5.1.2 Level 0 DFD



Level 0 DFD - Narrative

The preceding data flow diagram describes the level 0 of the proposed backstage administration system used by the platform managers to manage the platform. The entire backstage administration will break down into 6 subsystems based on their different functions.

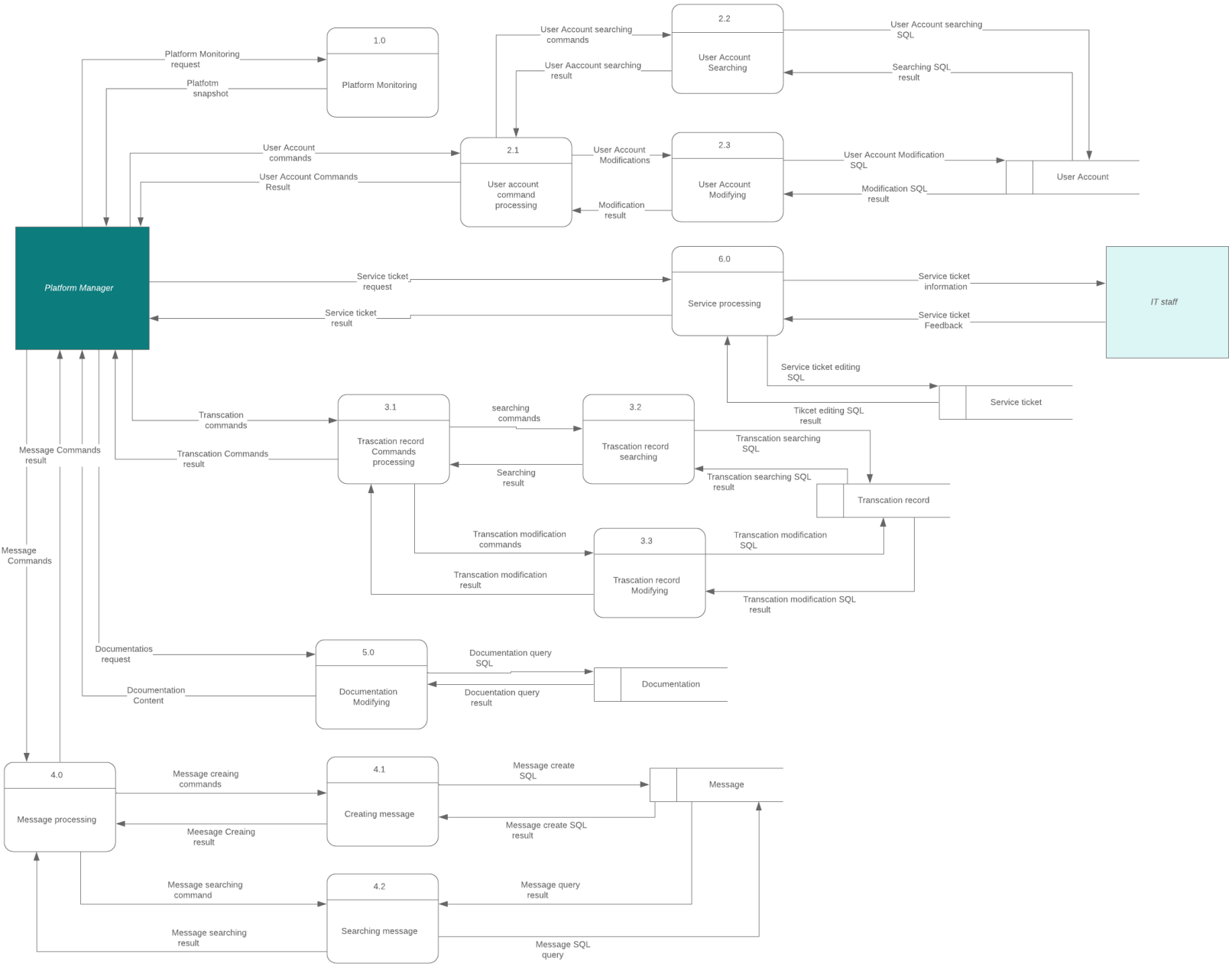
Platform managers:

1. Once the platform managers send commands to administration system, the command will be classified:
    a. If it is user account command, it will be sent to subsystem 2.0 to process the command.
    b. Once user account data processing completed, processing result will be sent back from subsystem 2.0 to platform manager.
    c. If the command is about transaction record, it will be sent to subsystem 3.0 to process the command.
    d. Once transaction data processing completed, processing result will be sent back from subsystem 3.0 to platform manger.
    e. If the command is about message center, it will be sent to subsystem 4.0 to process the command.
    f. Once message data processing completed, processing result will be sent back from subsystem 4.0 to platform manger.
    g. If the command is about platform documentation, it will be sent to subsystem 5.0 to process the command.
    h. Once documentation data processing completed, processing result will be sent back from subsystem 5.0 to platform manger.
    i. If it is system monitoring command, subsystem 1.0 will activated to capture the snapshot of real time system.
    j. The real time snapshot will be sent from subsystem 1.0 to platform manager.
    k. If the command is about initiating a service ticket, it will be sent to subsystem 6.0 to process.
    l. Subsystem 6.0 will sent back a response once the service ticket is updated by IT staff.

IT staff:

1. IT staff will receive ticket information from subsystem 6.0 after a service ticket initiated.
2. Once IT staff complete the service ticket, IT staff will send a ticket update back to the subsystem 6.0.

5.1.3 Level 1 DFD



Level 1 DFD-Narrative

The preceding data flow diagram describes the level 1 of the proposed backstage administration system. There are 6 subsystems in the proposed backstage administration system and every subsystem has 4 general functions: View, Add, Edit, and Delete. (Except subsystem 1.0 and 5.0).

Subsystem 2.0--User Account processing:

1. If the command is classified as “View”, the specific view request will be processed by function 2.2 and send to database to query the user account data.
2. Once function 2.2 complete the view processing, it will receive a data query result from user account database and send this result back to function 2.1 to finalize the result.
3. If the command is classified as “Add/Edit/Delete”, the modification command will be sent to function 2.3 to process.
4. Function 2,3 will send modification sql command to the database to modify the user account information. Once this process completed, a modification result will be sent back to function 2,1

Subsystem 3.0--Transaction Record processing:

1. If the command is classified as “View”, the specific view request will be processed by function 3.2 and send to database to query the transaction data.
2. Once function 3.2 complete the view processing, it will receive a data query result from transaction record database and send this result back to function 3.1 to finalize the result.
3. If the command is classified as “Add/Edit/Delete”, the modification command will be sent to function 3.3 to process.
4. Function 3,3 will send modification sql command to the database to modify the user account information. Once this process completed, a modification result will be sent back to function 3,1

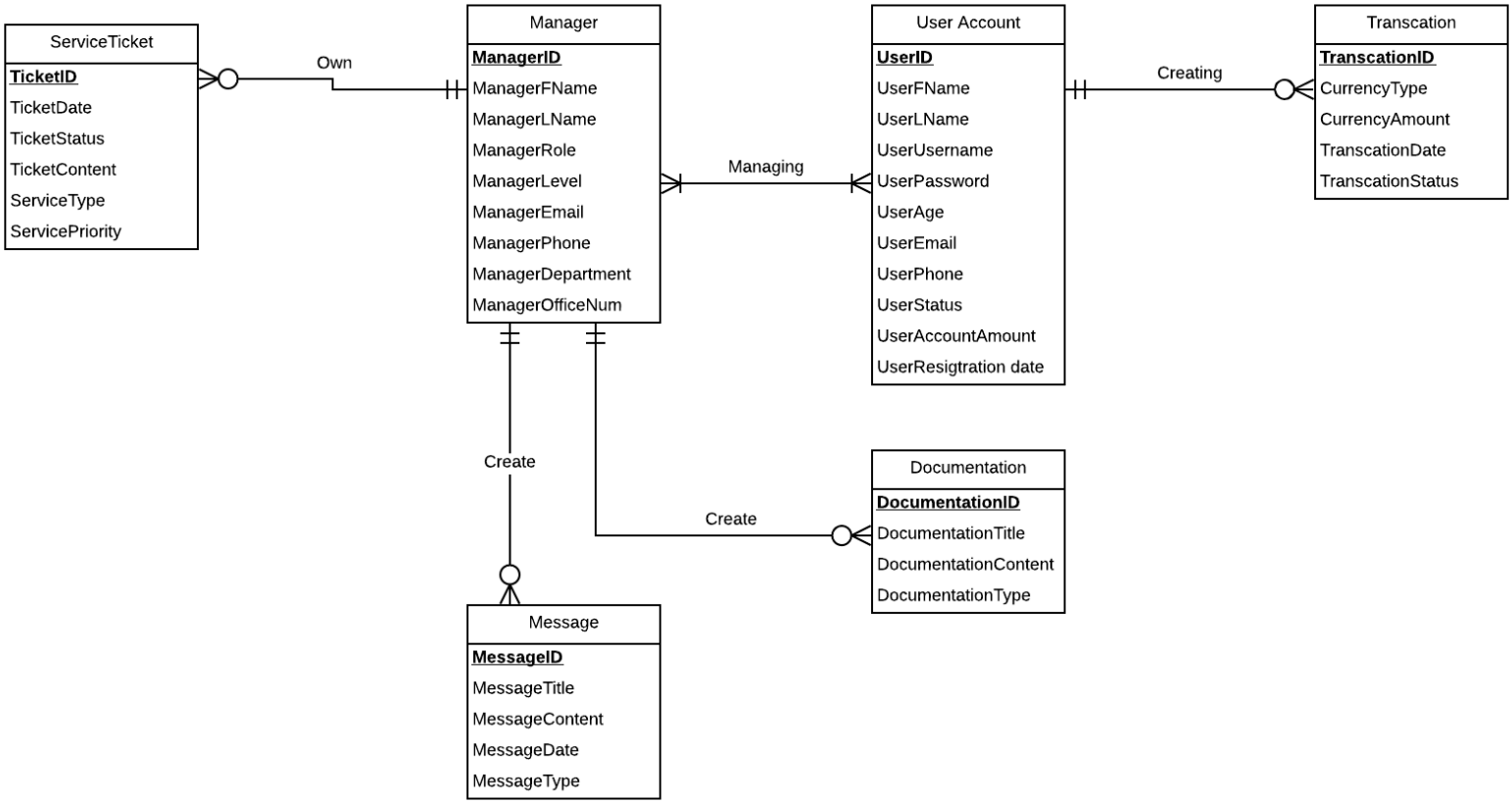
Subsystem 4.0--Message processing:

1. If the command is classified as “View”, the specific view request will be processed by function 4.1 and send to database to query the message data.
2. Once function 4.1 complete the view processing, it will receive a data query result from message database and send this result back to function 4.0 to finalize the result.
3. If the command is classified as “Add”, the message creating command will be sent to function 4.2 to process.
4. Function 4,2 will send create sql command to the database to modify the user account information. Once this process completed, a modification result will be sent back to function 4,0

Subsystem 5.0--Documentation Modifying:

1. If the platform manager wants to add/edit/delete platform documentations, manager will send a modification command to subsystem 5.0 to process it.
2. Once subsystem 5.0 receive the modification command, it will send a create/update/delete sql query to the documentation database.
3. Database will give a feedback to subsystem 5.0 about the modification result, once subsystem 5.0 received the result, a finalized response will be generated and send back to platform manager.

5.2.1 Entity Relationship Diagrams (ERD)-proposed system



5.2.2 ERD-proposed system- Narrative:

There are six entities in the proposed system ERD. The narratives of each entity are listed below:

Manager:

The entity stores platform manager account information including manager ID, manager Name, manager role, etc.

Relationship:

- A manager can manage one or many user accounts
- A manager can issue zero or many service tickets
- A manager can create zero or many messages
- A manager can create zero or many documentations

User Account:

The entity stores platform user account information including user ID, user Name, user password, etc.

Relationship:

- An user account can be managed by one or more platform managers
- An user account can create zero or many transactions

Transaction:

The entity stores platform transaction detail information including transaction ID, transaction date, currency type, etc.

Relationship:

- One transaction can be created by one and only one platform user account

Service Ticket:

The entity stores service tickets information including ticket ID, ticket date,ticket status, etc.

Relationship:

- One service ticket can be owned by one and only one platform manager

Message:

The entity stores platform messages information including message ID, Message content, message date, etc.

Relationship:

- A message can be created by one and only one platform manager

Documentation:

The entity stores platform documentation including documentation ID, documentation title, etc.

Relationship:

- A documentation can be created by one and only one platform manager

5.3 The Synchronized System Models

User Account	**User Account searching**	**User Account modifying**
UserID	**R**	**CRUD**
UserFName	**R**	**CRUD**
UserLName	**R**	**CRUD**
UserUsername		**D**
UserPassword		**UD**
UserAge	**R**	**R**
UserPhone	**R**	**R**
UserStatus	**R**	**RU**
UserAccountAmount		
UserResigtration Date	**R**	**R**

Transcation	**Transaction record searching**	**Transaction record modification**
TransactionID	**R**	**RD**
CurrencyType	**R**	**R**
CurrencyAmount	**R**	**R**
TranscationDate	**R**	**R**
TranscationStatus	**R**	**RUD**

Message	**Create Message**	**Searching Message**

MessageID	**CUD**	**R**
MessageTtle	**CUD**	**R**
MessageContent	**CUD**	**R**
MessageDate	**CUD**	**R**
MessageType	**CUD**	**R**

Documentation	**Documentation Modifying**
DocumentationID	**CRUD**
DocumentationTitle	**CRUD**
DocumentationContent	**CRUD**
DocumentationType	**CRUD**

ServiceTicket	**Service Processing**
TicketID	**CRUD**
TicketDate	**CRUD**
TicketStatus	**CRUD**
TicketContent	**CRUD**
ServiceType	**CRUD**
ServicePriority	**CRUD**

6. The Physical System Design

Based on the business requirements and analysis phase report, there are three candidate system solutions available for client to pick:

Candidate 1: Python Django Framework Web App---In-house development

Candidate 2: ASP .NET Framework Web App---In- house development

Candidate 3: Salesforce CRM platform---Software package

6.1 The Candidate System Solutions table and descriptions of candidates

Characteristics	**Candidate 1**	**Candidate 2**	**Candidate 3**
Portion of System Computerized	Membership Account and Transaction data management. System real time monitoring	Same as Candidate 1	Same as Candidate 1
Benefits	Python Django framework can be used to develop a small-middle size web app with high data processing performance in a short time	ASP .NET framework is align with the main Arena FinTech platform system. No API or other app needed to transfer the data	No system development needed because it is a purchased solution. The solution can be implemented quickly
Servers and Workstations	AWS ec2 instance, Linux Ubuntu workstation	AWS ec2 instance, MS Windows 10 workstation	No server needed, Windows/Mac OS workstation
Software Tools Needed	PyCharm, Postgre SQL, Django, HTML,	MS Visual Studio, MySQL Server, HTML, PHP	Apex API, Browser, SQL server studio
Application Software	**“**Professional**”** edition of PyCharm	**“**Enterprise**”** edition of MS Visual Studio	**“**Developer**”** edition of Salesforce
Method of data	Client/Server	Client/Server	Server

processing			
Input Devices and Implications	Keyboard & Mouse	Same as candidate 1	Same as candidate 1
Storage Devices and Implications	PostGre SQL server DBMS with 100GB arrayed capability	MySQL server DBMS with 100GB arrayed capability	MS SQL server DBMS with 100GB arrayed capability

Candidate 1 --- Description

Python Django Framework Web App

Description:

Candidate 1 is the first considered solution to solve the most urgent problem which is the lack of platform backstage administration system in Arena Fintech. After developing the customize in-house web app by using Python Django Framework, platform and manage the membership accounts and transaction record data in an efficiency way. They don't need IT staff to help them managing the platform.

Pros:

- Python Django framework can develop a small-middle website in a short time
- Python is better doing data processing than other program language
- Python Django framework is easy to maintenance
- Low Cost
- Short Development time

Cons:

- API app will needed to connect Python Django Framework Web App to the main system
- Performance will be limited if the web app scope increased
- High skilled IS people needed

Candidate 2 --- Description

ASP. Net Framework Web App

Description:

ASP. Net Framework Web App is the second considered solutions to solve the management issue in the Arena FinTech. Based on its MVC structure, mobile device management App can also be built for improving the management performance for Arena FinTech

Pros:

- Main platform system using the same framework, no API app needed
- Better performance in a large size web app
- Mobile app can be built
- Template available
- Low Cost

Cons:

- Long development lifecycle
- Complex structure
- High Skilled IS people needed

Candidate 3 --- Description

Salesforce CRM Platform

Description:

Salesforce CRM platform is the third considered solution to solve the backstage administration issues for Arena FinTech. Unlike the first two Candidates, salesforce CRM is a well-developed cloud solution provided by salesforce.com. It provide an efficient way for enterprise to manage and visualize data.

Pros:

- No system development needed
- Cloud service with 7/24 professional support
- Can be implemented in a very short time
- No high IS skilled people needed

Cons:

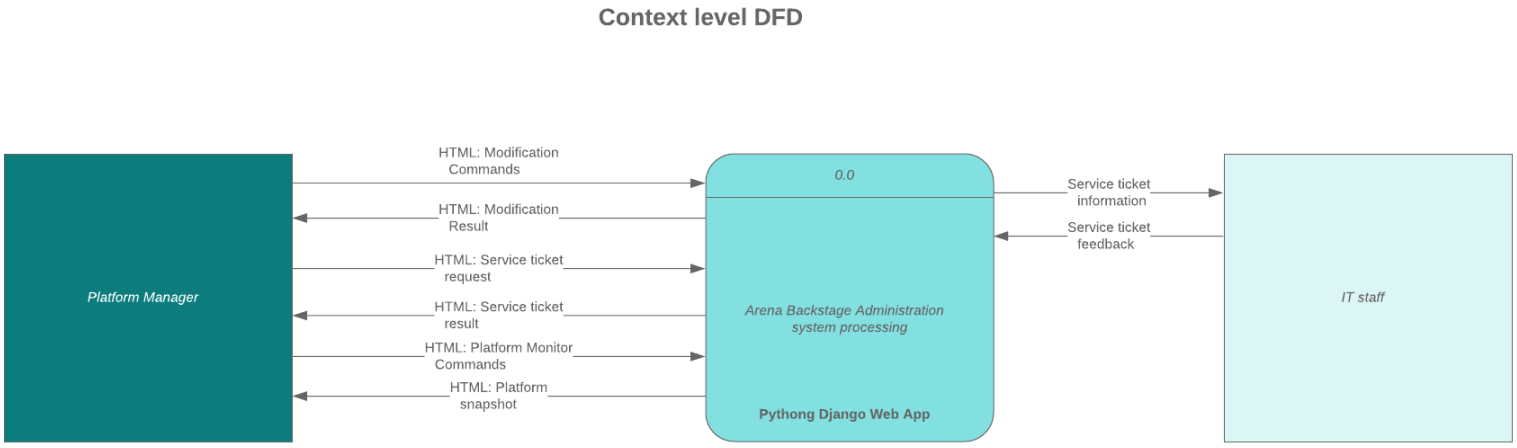
- Very Costly, the developer account is billed $150 per Month per user.
- Customize functions are limited

6.2 The Feasibility Analysis Matrix

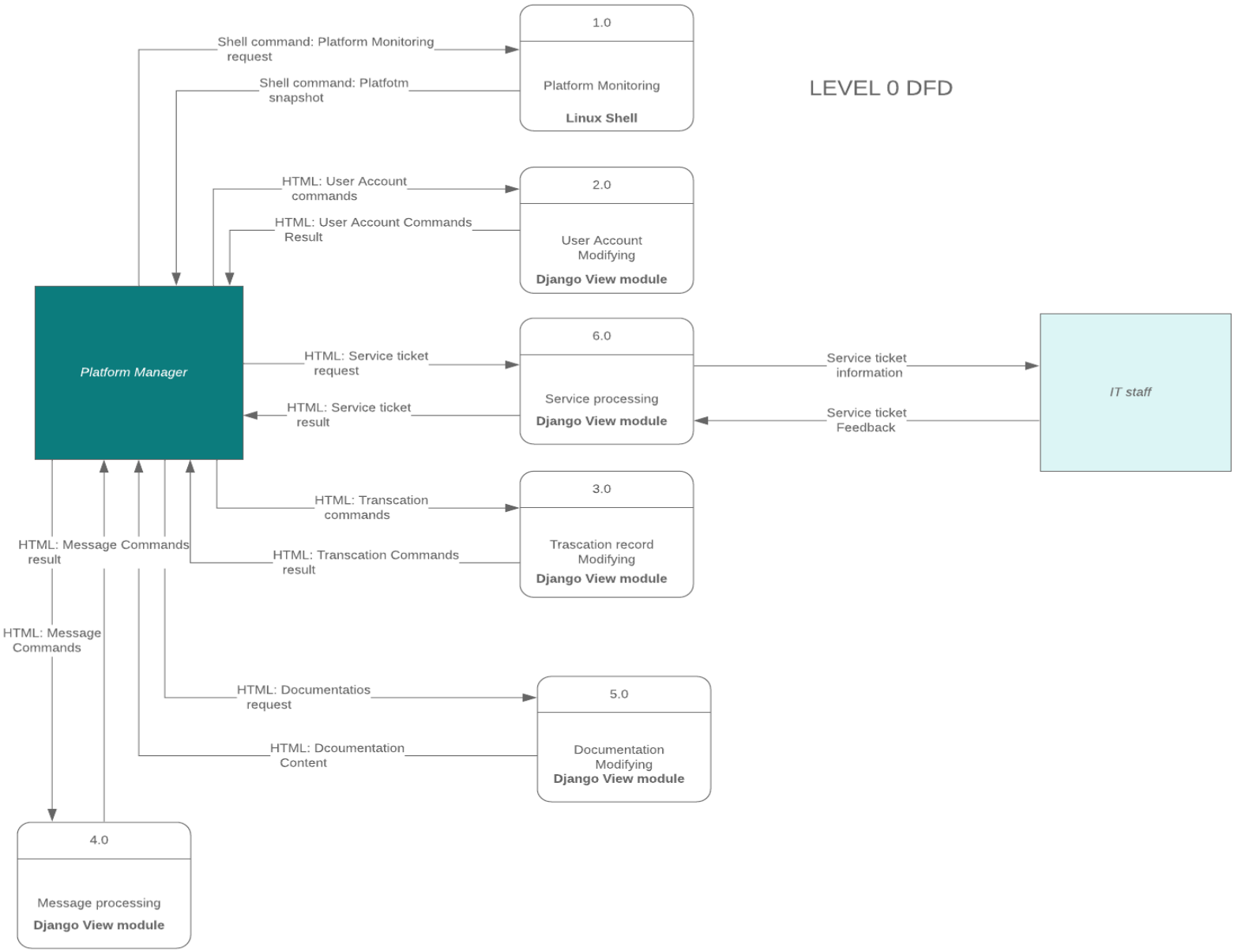
Feasibility Criteria	Weight	Candidate 1	Candidate 2	Candidate 3
Operational Feasibility	**30%**	Fully support user functionalities Score:100	Support some user functionalities Score:80	Customization will be limited Score:60
Technical Feasibility	**30%**	The company has never developed python Django before. Need to hire professionals or train employees Score:80	Current staff have experience working on ASP.NET platform. It will be easier for the company to program a ASP.NET websites. But certain functions cannot be applied with .NET Score:70	Current staff have limited experience in Salesforce. Need training. Developing customized functions will be challenging due limitation. Score:60
Economic Feasibility	**30%**	Cost to develop: Approximately 14,000. Score:100	Cost to develop: Approximately $16,000. Score:80	Cost to develop: Approximately $15,000. Extra Cost: 150 per person for Salesforce subscription. Score:70
Schedule Feasibility	**10%**	4 Month Score:80	Less than 4 month Score:80	Less than 3 month Score:100
Ranking	**100%**	**92**	**78**	**67**

6.3 The Physical DFDs

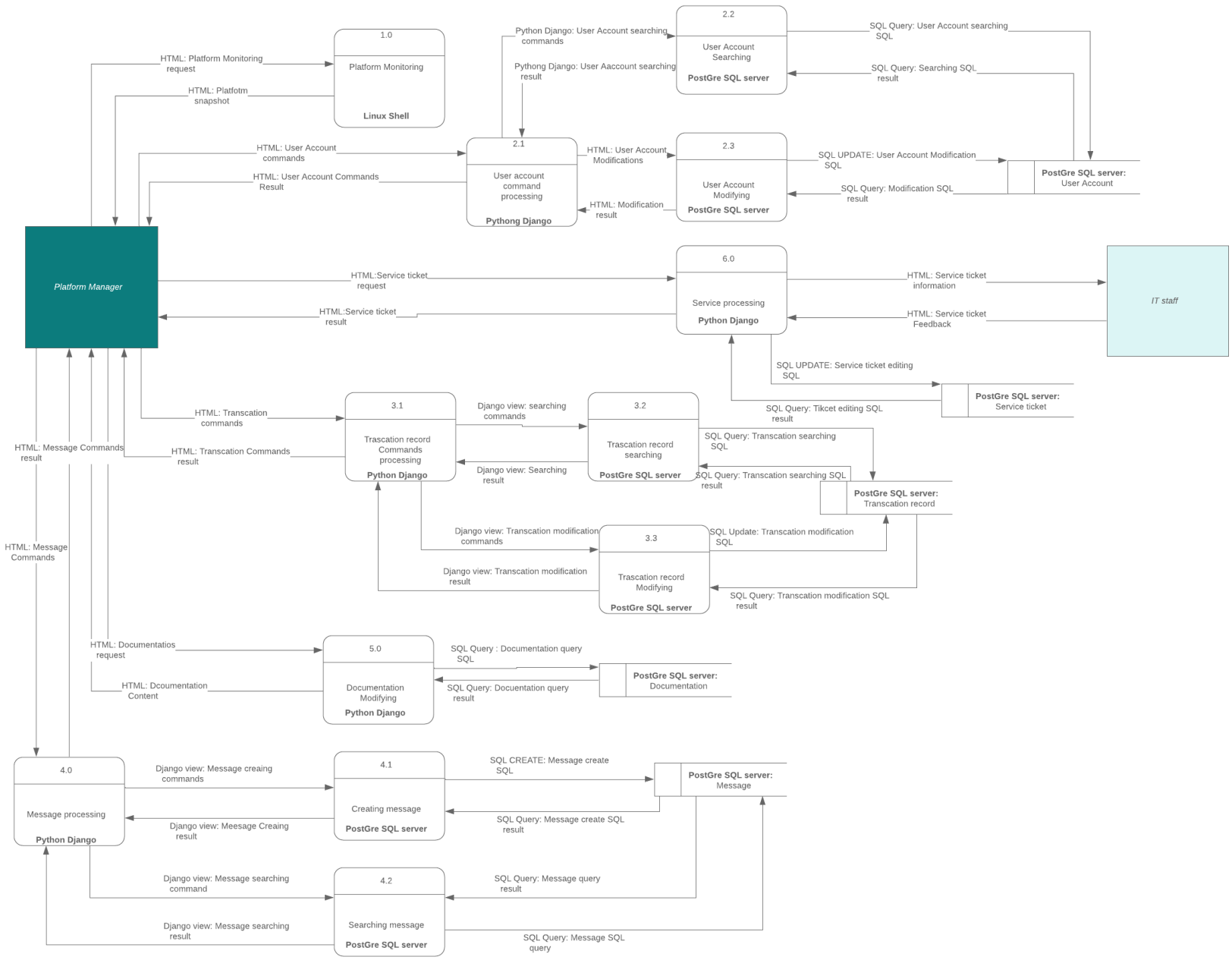
Physical DFD--Context level



Physical DFD -- level 0

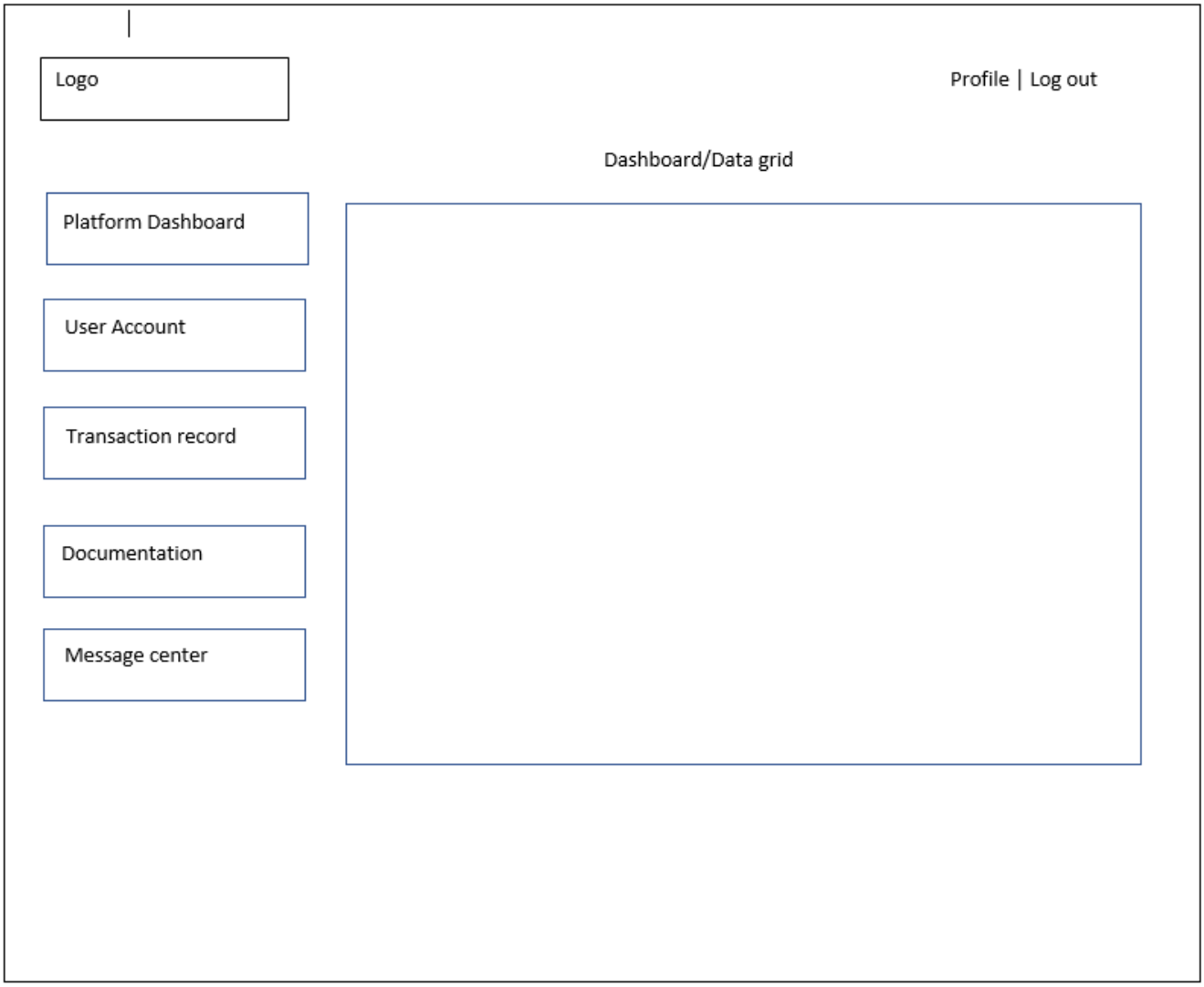


Physical DFD -- Level 1



6.4 Input and Output Design

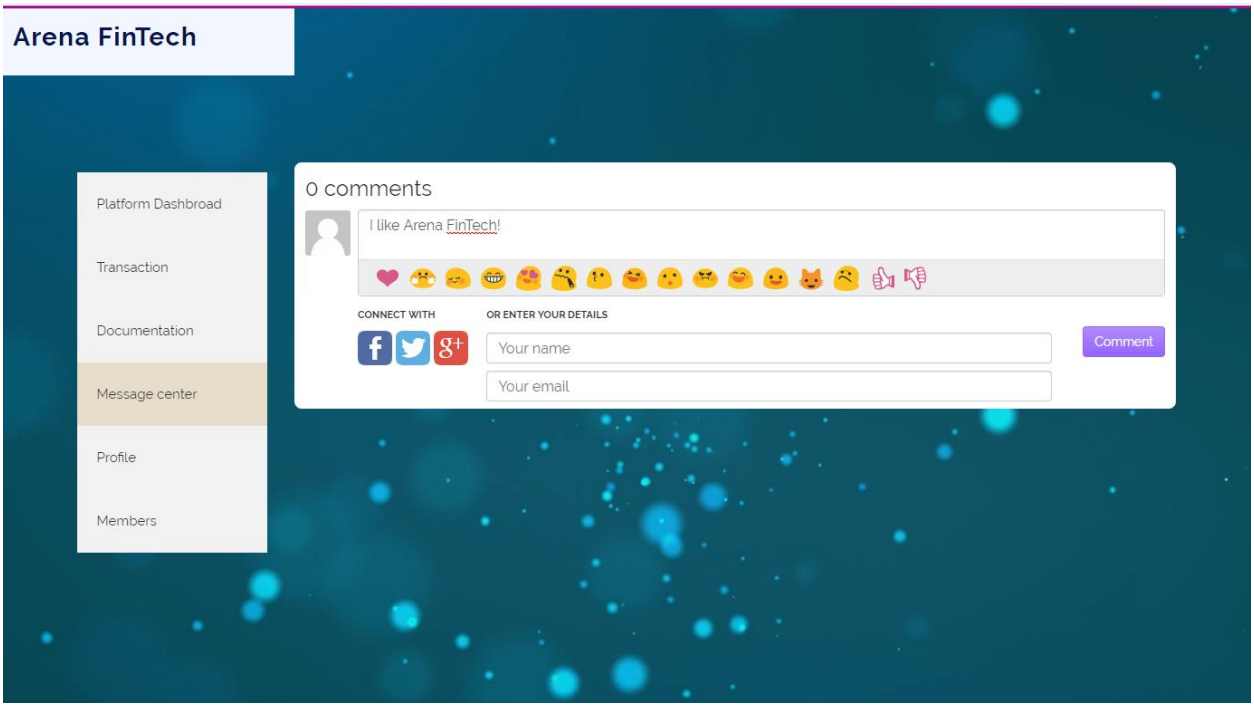
System Page Layout:



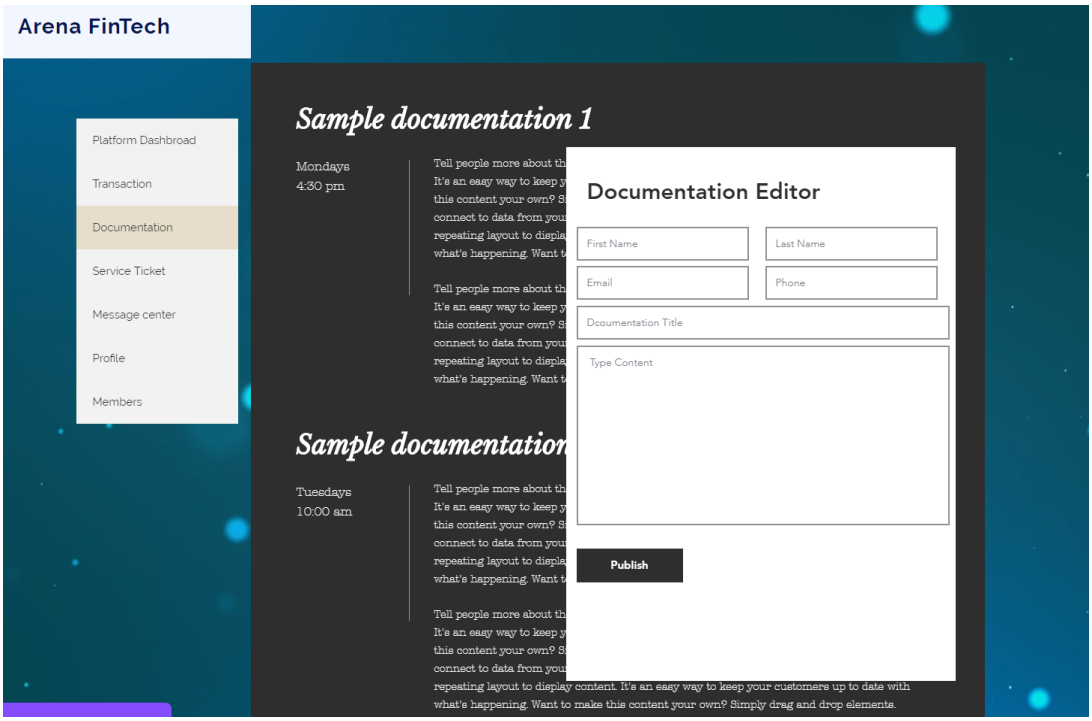
Input samples:

1) User account input sample

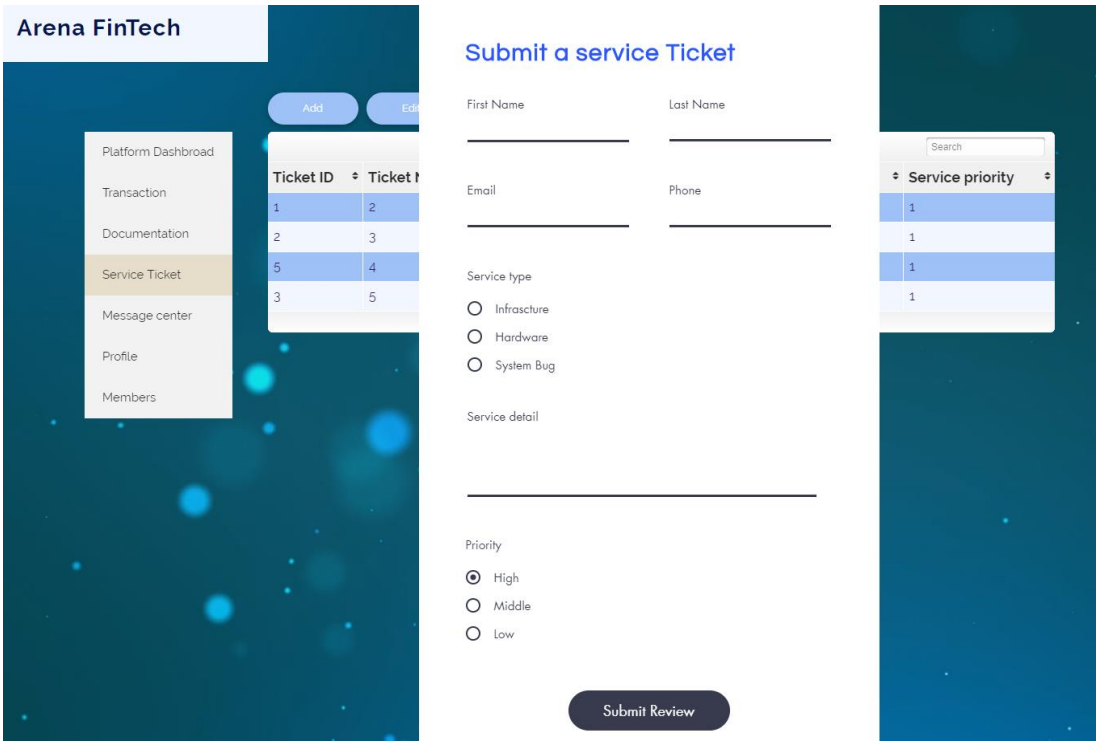
2) Message center input sample



3) Documentation input sample

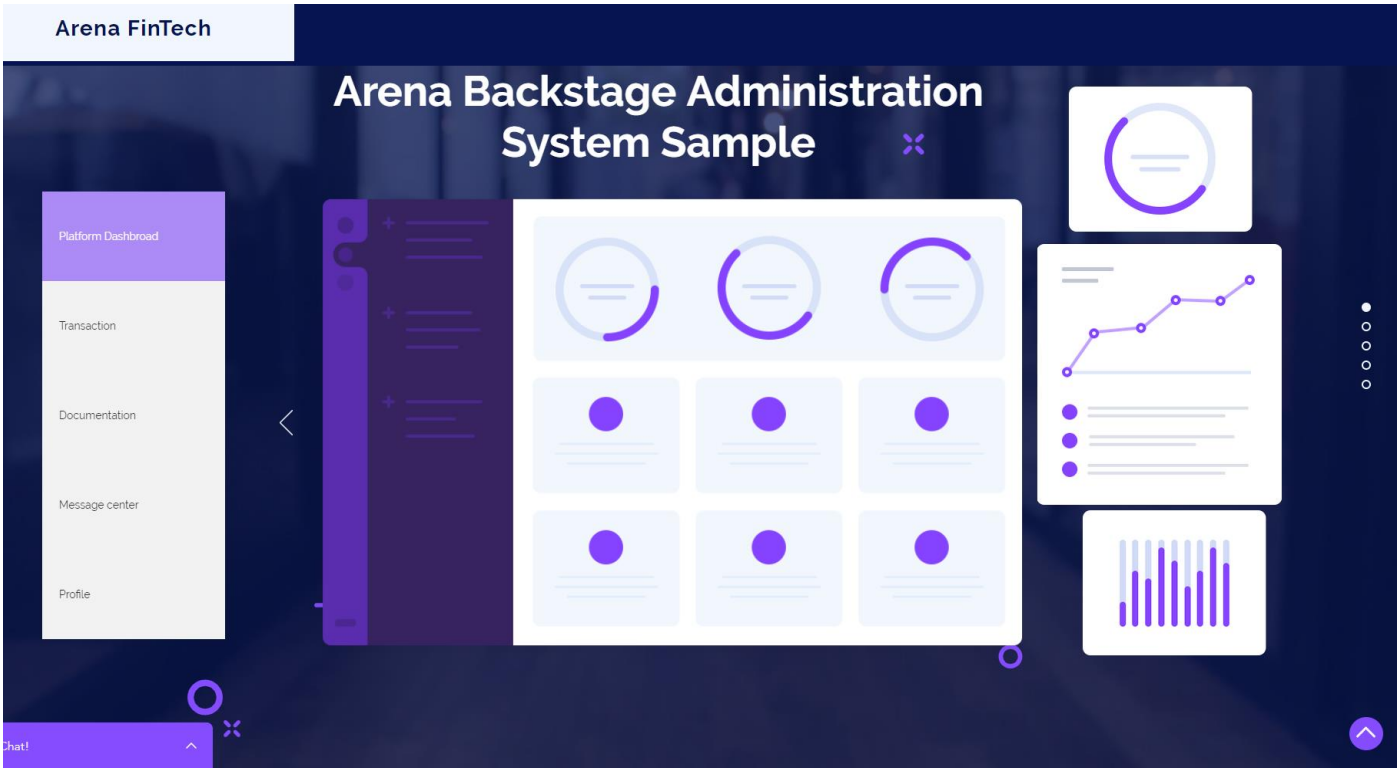


4) Service Ticket input sample

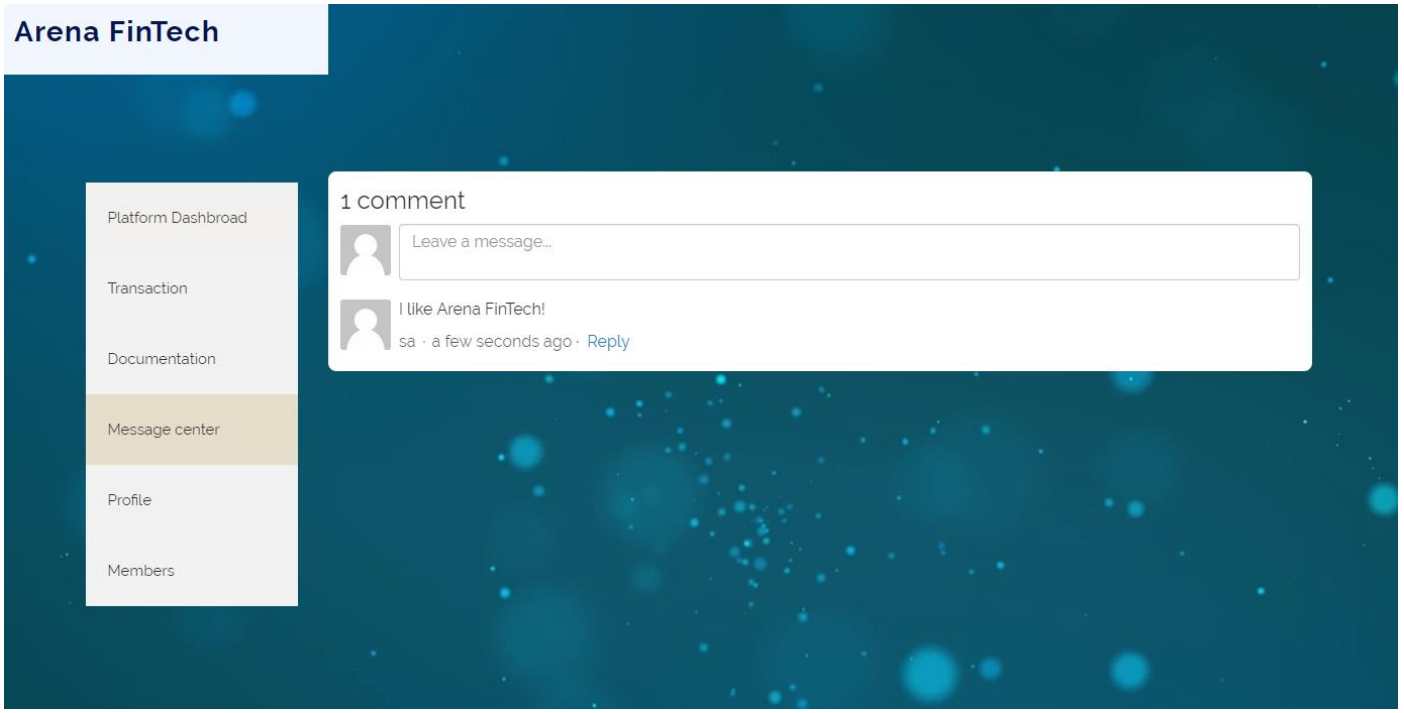


Output samples:

1) Platform Dashboard sample

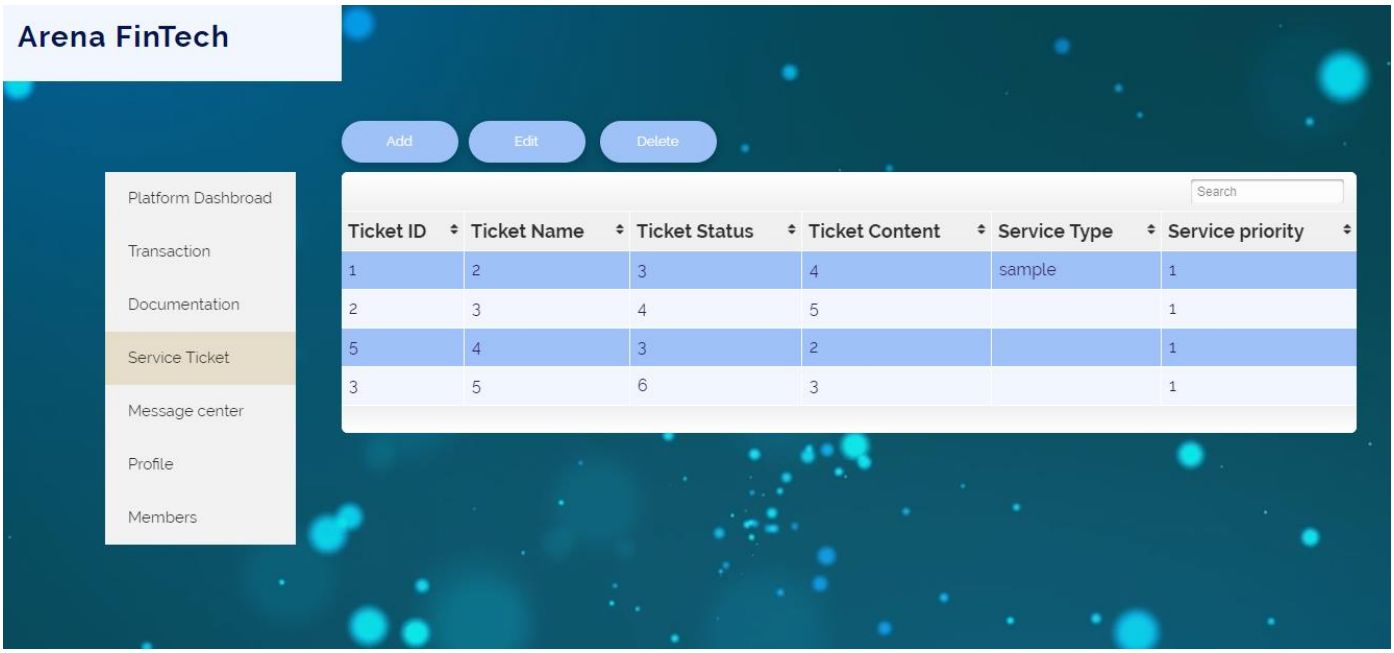


2) User account data grid sample



3) Message Center output sample

4) Service Ticket output sample



7. Implementation Plan

a. Implementation description

In order to secure a smooth and successful implementation procedure, Alpha consult team will work closely with our client and are open to their suggestions and requirements within the scope. The team divided the implementation plan into three phases. The first phase is information collection, the second phase is system designing and building and the third phase is system implementation.

Within the information collection phase, the team will have meetings with client to gather information about the current system including operations, limitations and pain points. The team will work collaboratively with IT department and IT manager to figure a feasible way to build and implement the system with certain time limitation. The second phase is system building. The team will provide detailed data flow diagram in context level, level 0 and level 1 to figure out a more efficient system operation, during which the team will also conduct testing and user acceptance. The third phase is the implementation of the system. The team will gather feedbacks and comments from the client and make final edits.

The team assumes the whole project duration will be 3 months and the team is determined the to finish the project before April,30 2019.

b. Implementation schedule

Project Implementation Plan			
Objectives: To develop the platform backend administration system			
Actions	**People Involved**	**Target Date**	**Comp. Date**
Collect current system information	Team, CFO, manager	2/11	2/8
Establish task force to review current procedure	Team	2/18	2/15
Discuss limitation and drawbacks with client	Team, IT staff, manager	2/18	2/15
Submit recommendations	Team	2/25	2/22
Redesign system	Team, IT staff	4/8	4/5
Comments by users and managers	Team, manager	4/12	4/12
Make final edits	Team	4/19	4.20
Documentations	Team	4/30	4/25
Distribute manual	Team	4/30	4/25
Project Completion Date: 4/30			

8. Lessons Learned

1. **Different requirements gathering methods.**

   Because our client is located in California, we spend the most of time communicating via phone at the very beginning of this project. Now, we have learnt various methods to gather information and requirements from our client. We would like to use interview or questionnaire via skype with our client.

2. **Input/design tools: Visio, etc.**

   As before, we only drew some scratches on paper to design the system and user interface. Now, we make use of several tools to do that such as Visio.

3. **Different methodologies.**

   In this project, we follow the traditional waterfall and went through the first three phases. We have learnt different methodologies for different natures of project. Now, we would like to use agile methodology to do a short duration project.

4. **A more detailed test plan between the candidate to better determine whether the candidate can accomplish the required task.**

   With a limited understanding of the technology. We weren't able to come up with a complete concept on how the testing process will be. Perhaps a more detailed learning in each candidate technology will help.

5. **Take resource limitation into consideration when developing the project.**
   The resource limitation should be further considered. Although we did run economic feasibility analysis. We did not take full consideration into the current resource that the company have. With a more detailed conversation with the client, we might do adjustment on the weight of feasibility analysis. This might create a different result.