

Software Requirements Specification

for

Fyre Flight

**Online Hotel & Flight
Reservation System**

Version 1.0

Prepared by Group 11

**Bryce Shaw
Matthew Vanderwey
Stephanie Walsh
Sean Watters**

ENSF 613 – Fall 2019

October 31, 2019

Table of Contents

1	Introduction.....	1
1.1	Purpose.....	1
1.2	Overview of Document	1
1.2.1	Document Conventions	1
1.2.2	Readers and Audience.....	1
1.3	References.....	2
1.4	Product Scope.....	2
2	Overall Description	4
2.1	Product Perspective.....	4
2.2	Product Features	4
2.3	User Classes and Characteristics.....	6
2.4	Operating Environment.....	6
2.5	System Constraints	7
2.6	User Documentation.....	7
2.7	Assumptions and Dependencies.....	7
3	System Features	8
3.1	System Function 1 – Search Function	8
3.1.1	Description of Function	8
3.1.2	Pre-conditions and promises.....	8
3.1.3	Possible Risks.....	9
3.1.4	Supplemental diagram [STD]	9
3.2	System Function 2 – Payment Function	10
3.2.1	Description of Function	10
3.2.2	Pre-conditions and promises.....	11
3.2.3	Possible Risks.....	11
3.2.4	Supplemental diagram [Decision Tree].....	11
3.3	System Function 3 – Administrate System.....	12
3.3.1	Description of Function	12
3.3.2	Pre-conditions and promises.....	12
3.3.3	Possible Risks.....	13
3.3.4	Supplemental diagram [Decision Tree].....	13
3.4	System Function 4 – Interface with Customer.....	14
3.4.1	Description of Function	14
3.4.2	Pre-conditions and promises.....	15
3.4.3	Possible Risks.....	15
3.4.4	Supplemental diagram [STD]	16

4	<i>External Interface Requirements</i>	17
4.1	<i>User Interfaces</i>	17
4.2	<i>Hardware Interface</i>	18
4.3	<i>Software Interfaces</i>	19
4.4	<i>Communications Interfaces</i>	19
5	<i>Non-functional Requirements/ Quality Requirements</i>	20
5.1	<i>Performance Requirements</i>	20
5.2	<i>Security Requirements</i>	20
5.3	<i>Portability Requirements</i>	20
6	<i>Prioritization and Release Plan</i>	21
6.1	<i>Product development prioritization</i>	21
6.2	<i>Development milestones</i>	21
6.3	<i>Feature Release List</i>	22
6.4	<i>Feature release plan</i>	22

List of Figures

<i>Figure 1-1: Fyre Flight Booking Request Execution</i>	3
<i>Figure 2-1: Level 0 Context Diagram</i>	5
<i>Figure 2-2: How Responsive Websites Work</i>	6
<i>Figure 3-1: Search Offerings Process</i>	8
<i>Figure 3-2: Search Offerings STD</i>	9
<i>Figure 3-3: Process Payment Process</i>	10
<i>Figure 3-4: Payment Function Decision Tree</i>	11
<i>Figure 3-5: Administrate System Process</i>	12
<i>Figure 3-6: Administrate Function Decision Tree</i>	13
<i>Figure 3-7: Interface with Customer</i>	14
<i>Figure 3-8: Interface with Customer STD</i>	16
<i>Figure 4-1: Client UI Template</i>	17
<i>Figure 4-2: Hardware Interface</i>	18

List of Appendices

Appendix A: Data Flow Diagrams

Appendix B: Entity Relationship Diagrams

Appendix C: Data Dictionary

Glossary:

Term	Description
AWS	Amazon Web Service
Domain	The area considered within the study
DFD	Data Flow Diagram. A diagram that shows flow of data and control in a software system
ERD	Entity Relationship Diagram. A diagram that shows the relationship among preserved data entity.
Data Store	A data-modelling notation to show the places that data rests.
OFHR	Online Hotel and Flight Reservation System.
STD	State Transition Diagram
UI	User Interface

1 Introduction

1.1 Purpose

This document is a Software Requirements Specification (SRS) that outlines the software requirements for Version 1.0 of an Online Flight and Hotel Reservation (OFHR) system known as Fyre Flight. This OFHR system includes the following key elements:

- A web interface where users can setup accounts, search for hotel and flight offerings, make bookings, submit payments and receive immediate confirmations for those bookings
- An interface with airlines and hotels (vendors) that pulls flight and hotel offerings in real-time, receives corporate data (payment terms etc.) from the vendors, and makes the user requested bookings directly with the vendors
- A backend system that aggregates online reviews of both the vendors and their offerings to optimize search results for the users
- A payment processing system that allows for direct bookings and immediate confirmations
- A login for system administrators to pull reports regarding the processed bookings
- An automated marketing system that analyzes what bookings are currently popular, user interests and external marketing data to drive sales with targeted advertisements

The primary intent of the proposed OFHR system detailed in this document is to provide end users (travellers) with current, relevant and highly rated travel options, which they can quickly and easily book directly through the system.

1.2 Overview of Document

1.2.1 Document Conventions

This document follows the standards for technical documents outlined in the charter for Group 11, including the established “Group 11” styles for Microsoft Word (Office 365) documents.

1.2.2 Readers and Audience

This document is intended to provide project managers, business analysts, vendors, system administrators, marketing staff, developers and testers with a clear and concise understanding of the proposed system’s features and modes of operation.

Section 2 of this document provides the context for the proposed system, including its major components and their external interfaces. This section also outlines the primary features of the system, anticipated user classes, operating environment, system constraints, user documentation, external dependencies and assumptions made in preparing this specification.

Section 3 presents detailed descriptions of the proposed system functions. This includes the required pre-conditions and promises for each function, as well as possible risks and supplemental diagrams (State Transition Diagrams, Decision Tables, Decision Trees) where applicable.

Section 4 outlines the requirements for external interfaces with the proposed system. This includes the proposed user, hardware, software and communication interfaces to be implemented. The logical characteristics of the user interface are described in this section; however, exact details of the user interface design will be included in a separate specification.

Section 5 reviews the non-functional requirements for the proposed system; including its performance, security and quality requirements. This includes the rationale behind these requirements to aid developers in understanding the intents so that they can make suitable design choices.

Section 6 describes the methodology used for prioritizing the functional requirements of the proposed system. Rankings for the system's functional requirements in order of priority, along with potential risks and proposed release dates, are also provided in this section.

Attached to this software specification document are four appendices which contain; Level 1 and Level 2 Data Flow Diagrams (DFD) for each main process, Entity Relationship Diagrams (ERD) for the proposed system, and a Data Dictionary.

Here are the suggested readings by audience:

Audience	Reading Suggestions	Sign-Off Required
Project Manager / Business Analyst	Sections 2 & 4 & 6	Yes
Vendor	Sections 2 to 4	No
System Admin	Sections 2 to 4	Yes
Marketing Staff	Sections 2 & 6	No
Developers	Sections 2 to 6	Yes
Testers	Sections 2 to 5	No

1.3 References

The following references were used in developing this document:

1. J. Rule and B. McFarland, *Fyre Flight: Vision & Scope*, Business Case, 2017. [Tweet] Since taken down from Twitter before it could be used as evidence in FBI investigation.
2. U.S Department of Health and Human Services, "User Interface Design Basics", 2017. [Online]. Available: <https://www.usability.gov/what-and-why/user-interface-design.html> [Accessed October 26, 2019]
3. OWASP™ Foundation, "Web Standards and Specifications", 2016. [Online]. Available: https://www.owasp.org/index.php/Web_Standards_and_Specifications#References [Accessed October 26, 2019].
4. Storyblocks Products and Engineering, "Web Architecture 101" [Online]. Available: <https://engineering.storyblocks.com/web-architecture-101-a3224e126947>

1.4 Product Scope

The proposed system will allow prospective travellers (or their agents) to access currently available flight and hotel offerings, on their desktop or mobile device, from highly rated vendors for destinations all over the world. The prospective traveller will be able to search for flights from their origin city to their destination city, as well as hotels in any city of their choice. Users will be

able to search for flights based on travel origin, destination, dates, class, preferred airline and sort results based on price or duration of travel. They will also be able to search for hotels by brand, stars, average user ratings and available amenities. Search results provided to prospectively travellers will be optimized based on preferences set by the user in their account, what is currently popular with other users and ratings provided by third parties such as Trip Advisor™. This same information will be used to power an advanced algorithm for targeting users with direct digital advertisements, creating a positive feedback loop to drive sales.

On the vendor side, hotels and airlines will be able to send available offerings to the system for quick sale. With secure payment integration, the system guarantees vendors direct payment for their offerings and travellers immediate booking confirmation.

The system will also provide system administrators with tools for monitoring system performance and sales. Allowing for in-depth business and market analysis.

Figure 1-1 below show what the typical booking request execution would look like in the proposed system:

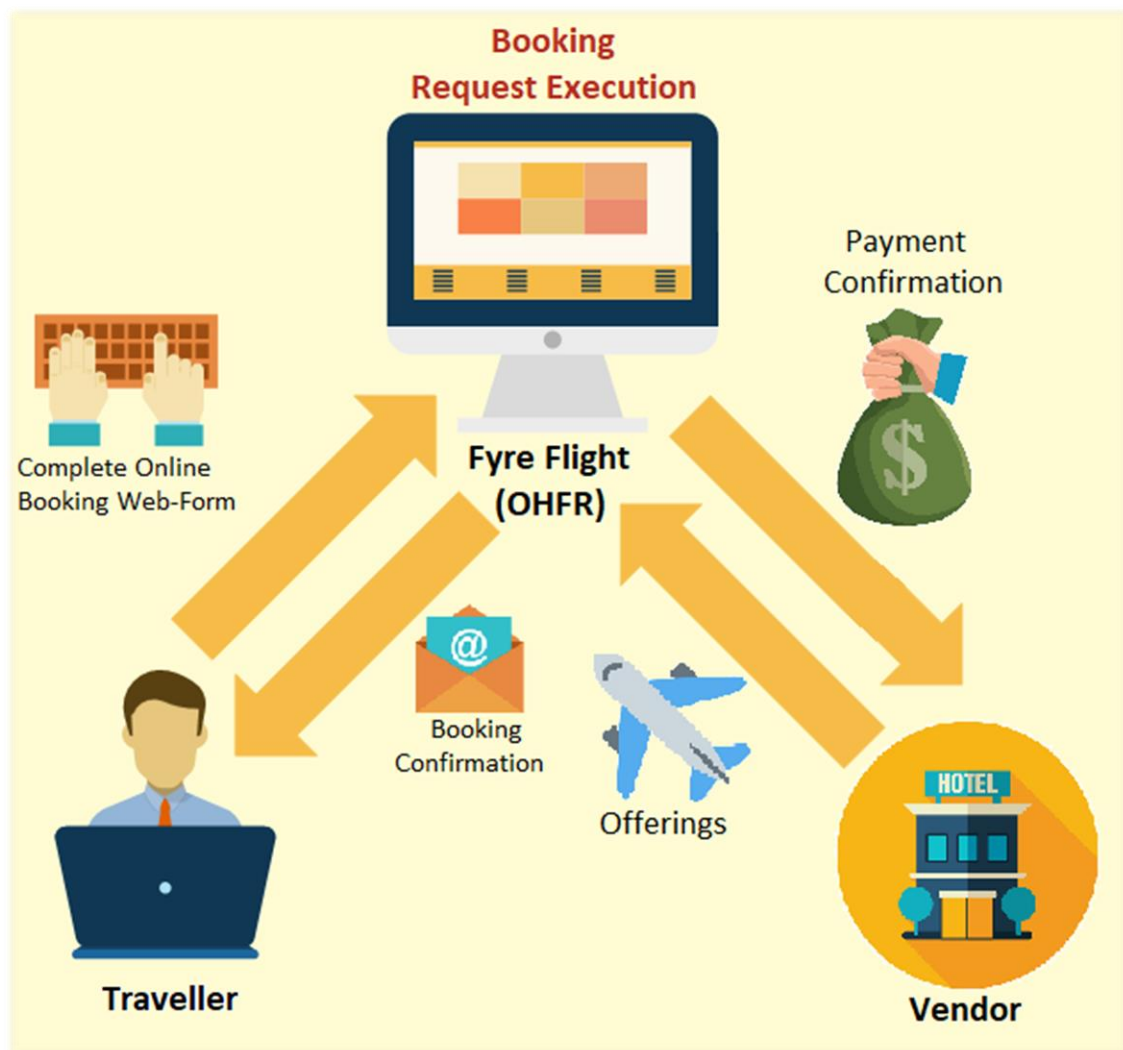


Figure 1-1: Fyre Flight Booking Request Execution

Original source: <https://conceptdraw.com/a3164c3/preview>

2 Overall Description

This section gives background information about the online flight and hotel reservation system. While this section will not describe the components in detail, it will provide high level details about the system components.

2.1 Product Perspective

This product is a new, web-based travel booking site focusing on hotel and flight reservations. The whole system connects the front end (user interface) with the back end: vendor interface, banking systems, and reporting systems. Product features and system requirements are outlined through the rest of this document.

2.2 Product Features

This software product is intended to be used by anyone who wants to look for flights and/or hotels, whether they are a travel agent helping to book a large group or an average person planning a trip for themselves. The major functions of the software include:

- An online user interface where the user can search for hotels and flights for anywhere that our vendors fly or reside;
- A system where airline and hotel vendors can provide their available offerings to be able to be booked through our system;
- A connection with banking vendors that can process payments for bookings; and
- A backend system that allows the system administrator to access data to be able to pull reports and create advertising to drive sales for their site.

The high-level context diagram is shown below in Figure 2-1.

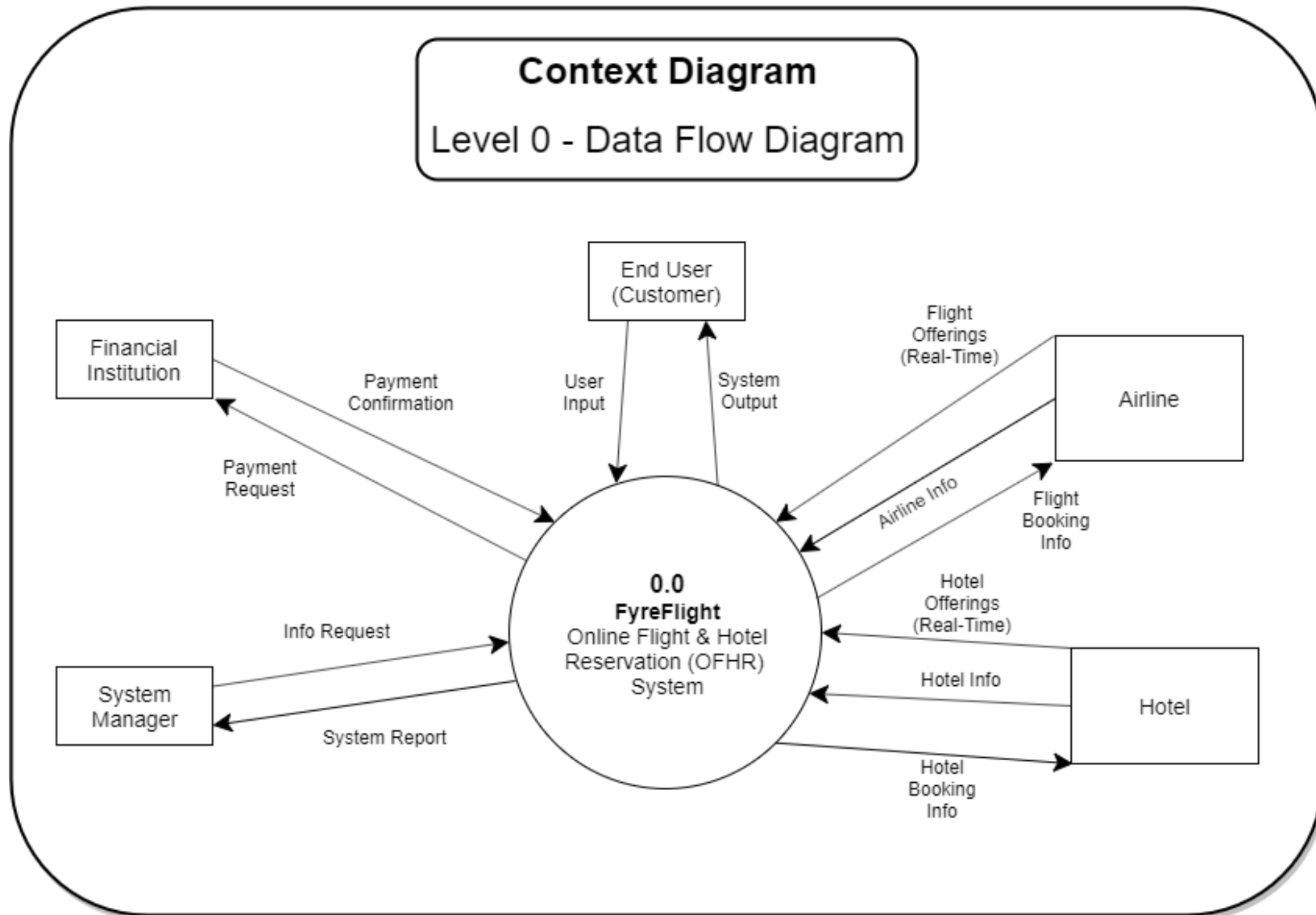


Figure 2-1: Level 0 Context Diagram

2.3 User Classes and Characteristics

On the front end, the primary user of this web-based software will be the everyday traveller. Their use will be limited to searching, booking, and paying for their travel. This is the most important class of user to satisfy and upon which the market viability of the entire system depends. Travellers (front end users) will have the lowest level of access, but also require the highest level of refinement and performance for their interface. As users will have no access to the inner workings of the system, they will be free to create, modify and/or delete their own user accounts.

On the back end, the main user will be the system administrator who can access reports on user's bookings and system status and provide direction to the drive sales process. This class of user will have the highest level of access privilege within the system.

Ranked between the traveller and the system administrator, in terms of importance to satisfy, we have the vendor class of user. The vendor interface with the system is primarily automated, but there will be times when manual vendor input will be required, such as when they first sign on to the system. This class of user's access to the system should be strictly controlled by the system administrator, requiring the administrator's approval for both adding vendors to the system and for any changes to their corporate information in the system.

2.4 Operating Environment

This will be a web-based application where users will access the site from their desktop, laptop, or mobile device through their chosen web browser. The system will be hosted on a server (or servers) which will be the responsibility of the system administrator to manage directly or coordinate the management of with the hosting provider. All system processing will happen on the hosting server and no processing will be offloaded to any user system interfacing with this system. Developers are required to follow the Web Standards and Specifications outlined by the OWASP™ Foundation [3] in terms of web-based performance and system security. A graphic of how responsive websites, such as this one, should work is shown below in Figure 2-2:



Figure 2-2: How Responsive Websites Work

Original source: <https://www.micsupport.com/blog/responsive-website-work/>

2.5 System Constraints

Because the final product will be web based, it must be fast enough on both desktop and mobile devices to keep the user engaged. The system must work on all types of web browsers and have similar access speeds for each platform. This may limit the options available for final design purposes.

The system must talk with the airline/hotel interface to provide real-time data to the users. This may pose a problem if this interface is not designed properly with a possibility of a glitch between the vendor interface and the user interface. This connection must be both fast and reliable.

The system will also contain a user information database that must be secure. Users should not be able to see other users' information.

This system will also have a payment processing system. The developers must research available options and pick one that has high security ratings (agreed upon by the client), like PayPal for example.

The site must also have a security certificate and be secure with a https:// URL.

2.6 User Documentation

There will be an FAQ section on the site as well as contact information for the user to access any help they need with booking their travel. System administrator and vendors must also be provided information packages outlining their responsibilities for working with the system. This information should be provided in a highly searchable and easy to understand format, in a secure location on the server to which only any one class of user has access to their own documentation, aside from the system administrator who should have access to all online documentation.

2.7 Assumptions and Dependencies

For the purpose of this software, we are not developing a payment processing software, but making use of API connections to well-known software that is available in the marketplace. We are assuming that this software provides secure payment for the user and does not store their information without their consent.

The function of this system, as specified herein, is highly dependent on constant communication with vendors. Both in terms of receiving real-time up-to-date offerings and immediate booking confirmations, as well as being able to provide secure payment confirmations. Secure, fast and reliable web hosting will be critical for this system effective operation.

These specifications do not call for the use of any proprietary sub-systems, with the intention of making the entire system as cost-effective as possible to develop and maintain.

3 System Features

This section outlines some of the major system functions of the OFHR system.

3.1 System Function 1 – Search Function

3.1.1 Description of Function

- a. Airline Search – User submits requested airport code origin and destination, and a date range
- b. Hotel Search – User submits requested city, distance from landmarks, price range, star range, and date range

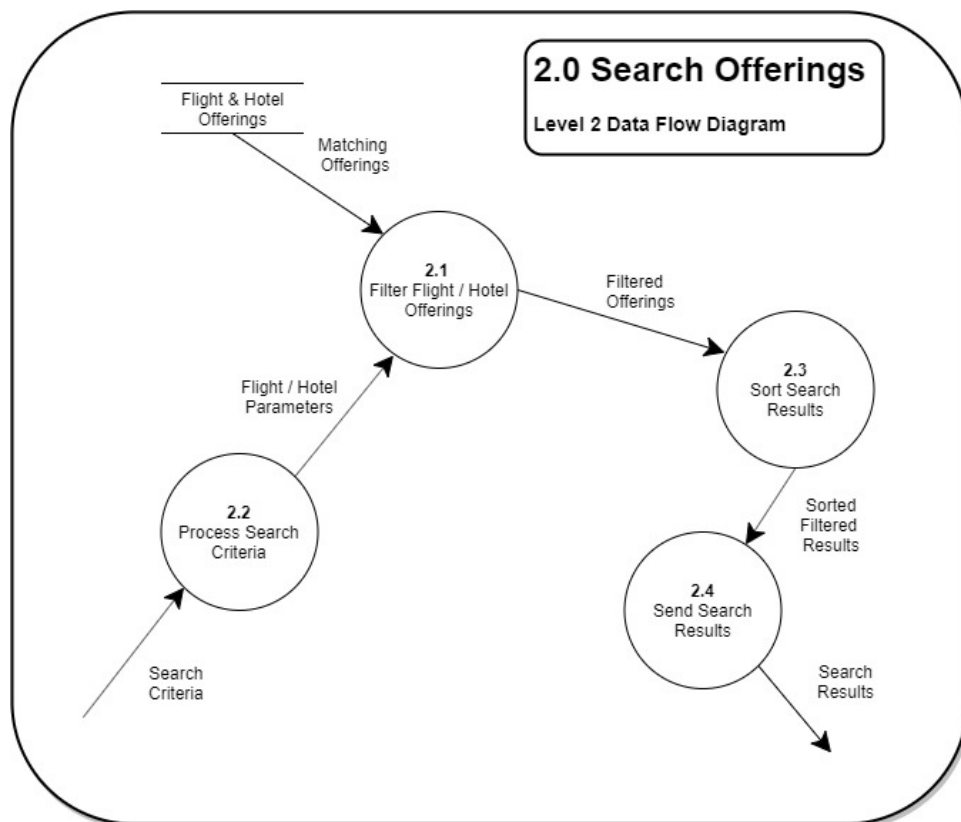


Figure 3-1: Search Offerings Process

3.1.2 Pre-conditions and promises

Process: Searching for flight and/or hotel

Preconditions:

- User inputs valid information
- Information is correct from the hotel/airline companies

Post Conditions:

- **If** valid information is entered
 - **If** search related results found in airline/hotel offerings
 - Format the searched offerings
 - Display the searched offerings
 - **Else**
 - Display message
 - Save issue in database for admin to review for reasons why
- **Else**
 - Display message

3.1.3 Possible Risks

- Two users are using the app at the same time, booking the same flight at the same time, with the flight only having one seat left on it.

3.1.4 Supplemental diagram [STD]

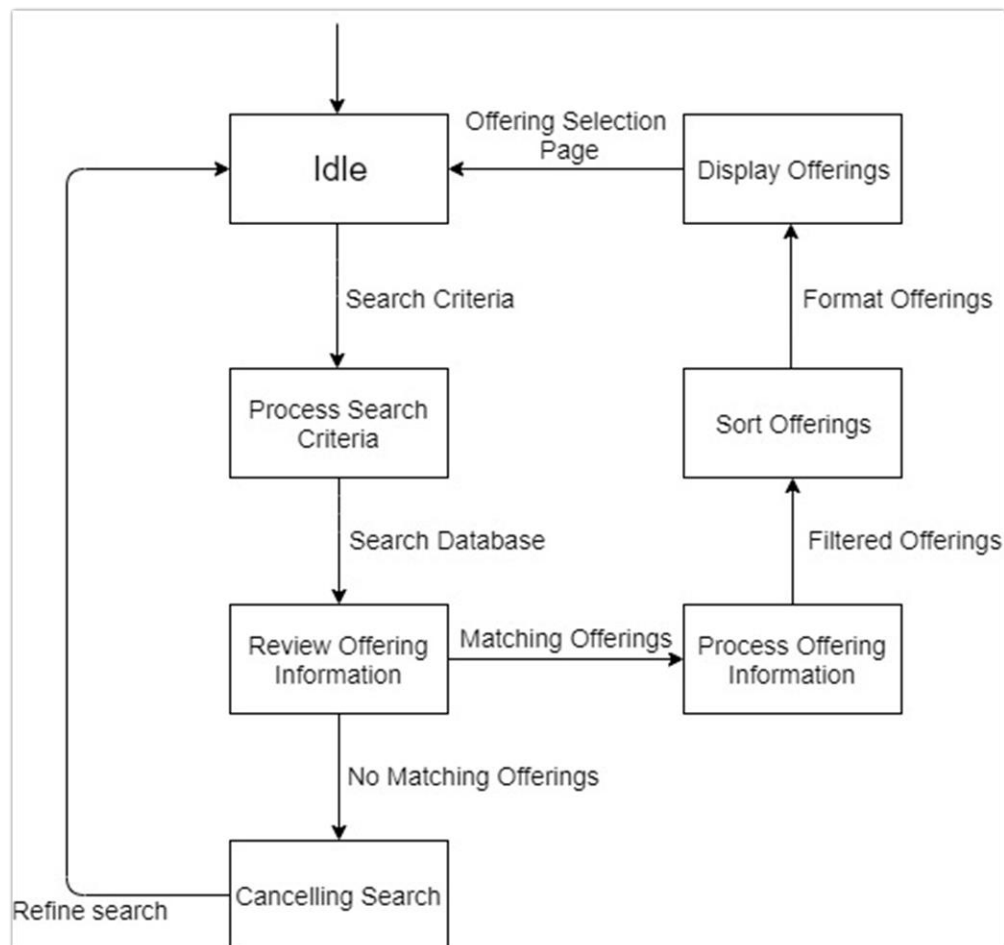


Figure 3-2: Search Offerings STD

Table 1: Decision Table for Searching Flights

Valid Date?	Y	Y	Y	Y	N	N	N	N
Origin Airport Code?	Y	N	Y	N	Y	N	Y	N
Destination Airport Code?	Y	Y	N	N	Y	Y	N	N
Display Offerings?	Y							
Prompt User for more Information?		Y	Y	Y	Y	Y	Y	Y

3.2 System Function 2 – Payment Function

3.2.1 Description of Function

Upon checkout the user inputs their credit card information to pay for services provided. The information is sent to the bank, which processes it and sends back a confirmation when/if validated.

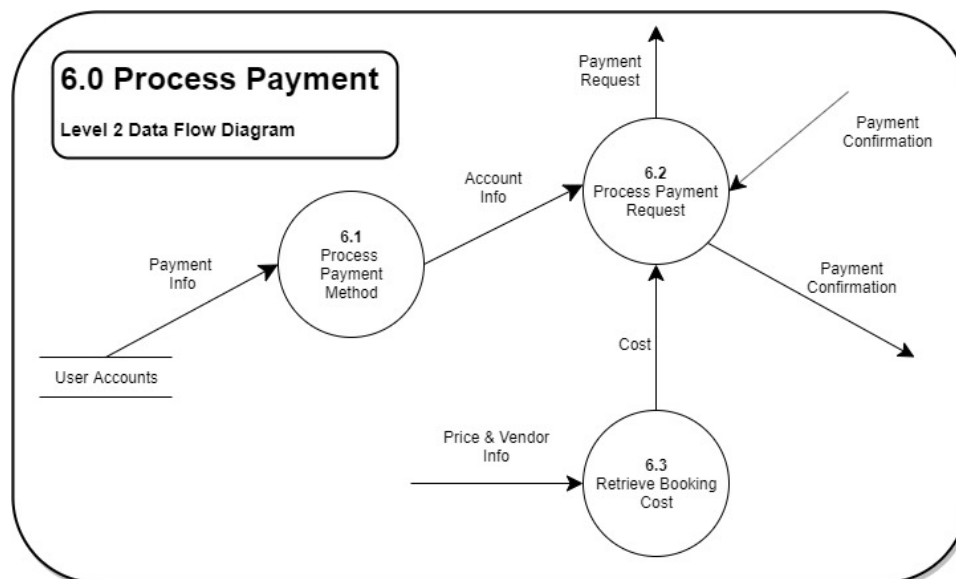


Figure 3-3: Process Payment Process

3.2.2 Pre-conditions and promises

Process: Paying for flight and/or hotel

Preconditions:

- User needs valid credit card
- Valid flight and hotel have been selected for purchase

Post Conditions:

- **If** valid order
 - **If** valid credit card
 - Process order
 - Save order to database
 - Save credit card info to user account
- **Else**
 - Display message

3.2.3 Possible Risks

- Credit card information from the user getting intercepted by hackers

3.2.4 Supplemental diagram [Decision Tree]

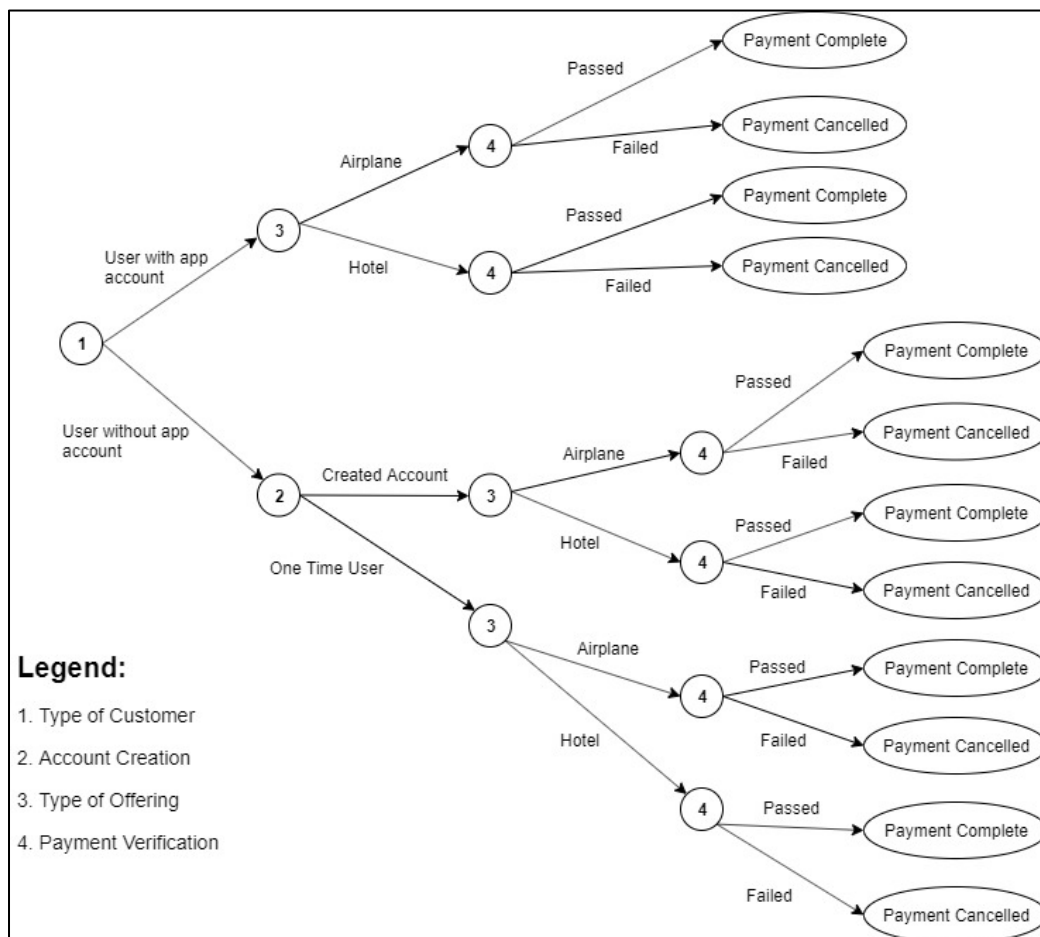


Figure 3-4: Payment Function Decision Tree

3.3 System Function 3 – Administrative System

3.3.1 Description of Function

As offerings are searched by users and booked, the data is stored within a database to allow for the corporate managers to review the data for workflow related improvements, UI improvements, and customer relation improvements. Data is also compartmentalized as some will be useful to the marketing side of the business and some to the corporate side. The managers will be able to search any and all data for specific or general values in order to visualize the businesses standings.

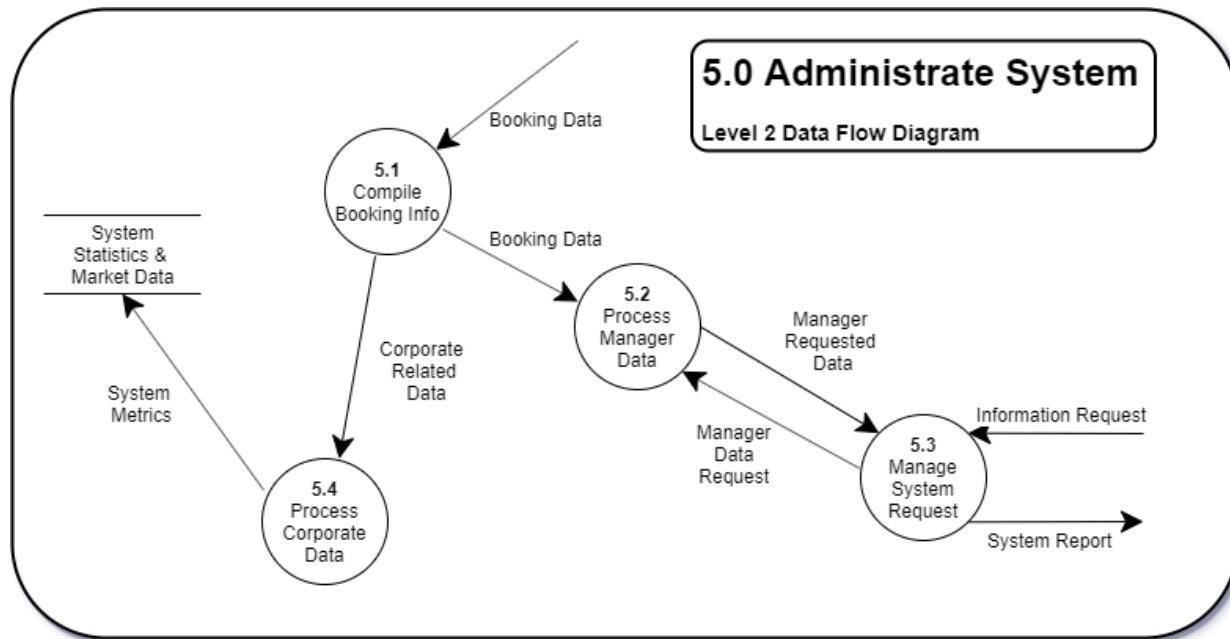


Figure 3-5: Administrative System Process

3.3.2 Pre-conditions and promises

Process: Collect and distribute business data to necessary business areas

Preconditions:

- The requested data is be collected and stored in the database

Post Conditions:

- **If** data is found in database
 - **If** requested from manager
 - Filter the related data
 - Format requested data
 - **Else If** requested from specific corporate department (ie. Advertising)
 - Filter related data
 - Format related data
 - Store 'System Statistics and Market Data' in database
- **Else**
 - Display message

3.3.3 Possible Risks

- All or some of the data gets purged from the data base and lost forever
- Certain necessary data is not being collected
- The wrong data or unnecessary data is being stored in either the booking data database or the System Metrics and Statistics database therefore taking up space within the database
- Requested data is interrupted by hackers and stolen for malicious use

3.3.4 Supplemental diagram [Decision Tree]

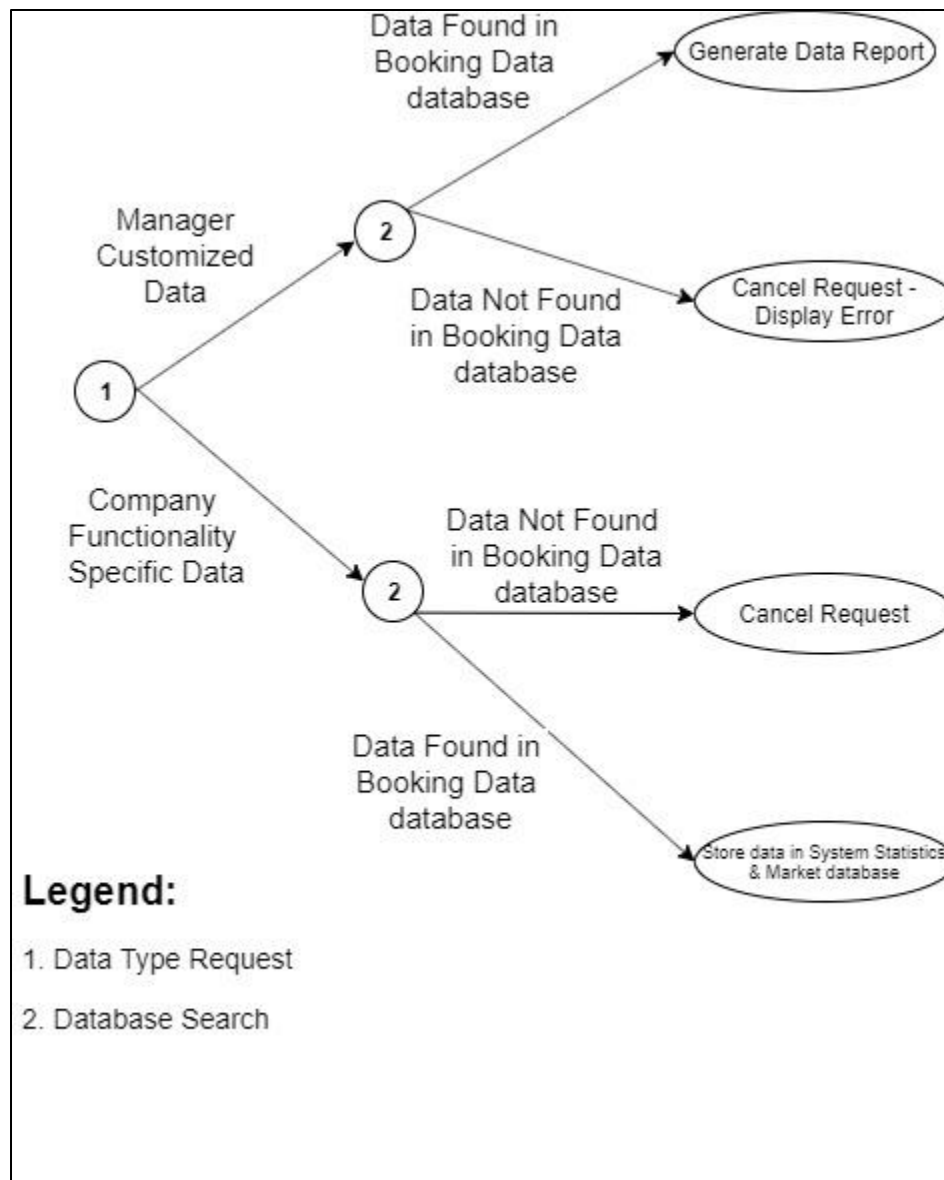


Figure 3-6: Administrative Function Decision Tree

3.4 System Function 4 – Interface with Customer

3.4.1 Description of Function

When a user interacts with the application, the user will have numerous features at their disposal, including searching and booking of both hotels and flights. The user will make an account in which the application will store the necessary information in order to facilitate the booking of the offerings provided by the vendors. User search and booking information will be stored to help with future improvements to the application and provide user specific advertisements and promotions based on their history on the application

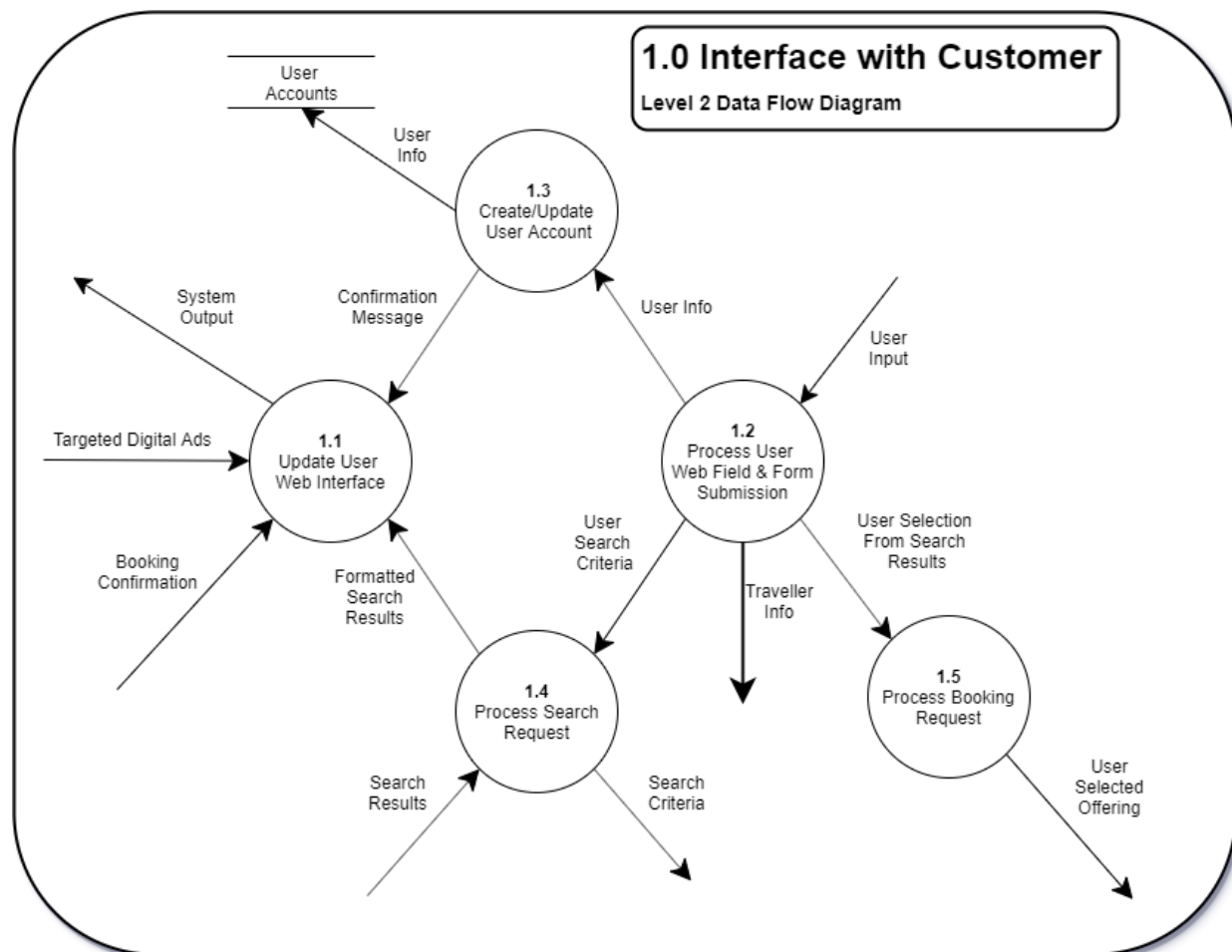


Figure 3-7: Interface with Customer

3.4.2 Pre-conditions and promises

Process: The main user interface for the customer using the application

Preconditions:

- The user visits the site
- The user utilizes the application to search and/or book offerings

Post Conditions:

- **While** customer is viewing the website
 - Display advertisements based on cookie data
 - **If** user has account
 - Display advertisements and promotions based on previous history
 - **If** search feature is utilized
 - Proceed to Search Offerings function (Sec. 4.1)
 - **If** Offering chosen to book
 - Display and Process booking forms
 - Proceed to Manage Booking function
 - **Else** If user creates account
 - Prompt user to fill in necessary information fields
 - Store data in 'User Accounts' database

3.4.3 Possible Risks

- User Interface has too many features and becomes overwhelming for the user
- User Interface is unorganized and confusing for the user
- Too many advertisements displayed per page
- User data isn't properly stored

3.4.4 Supplemental diagram [STD]

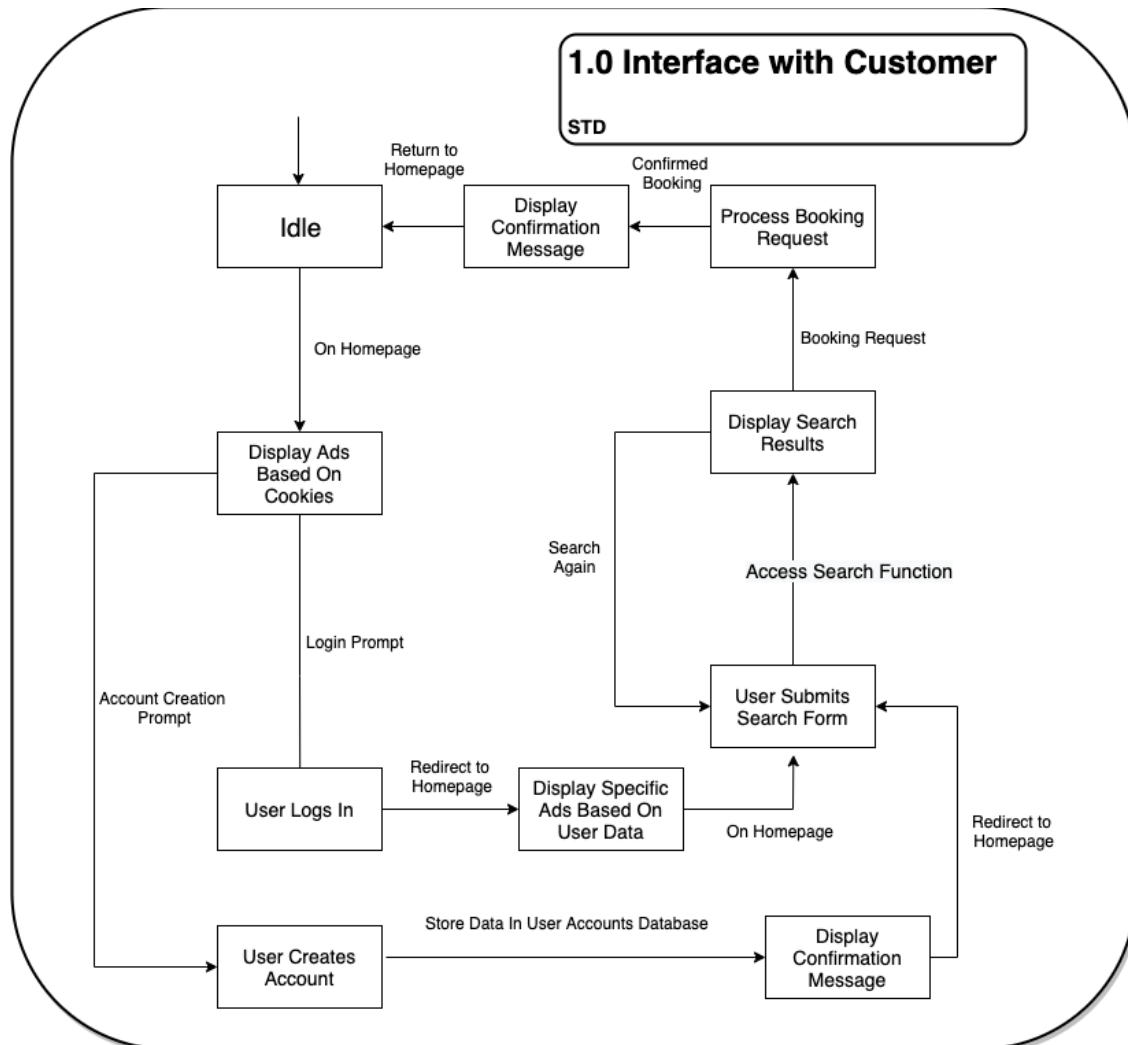


Figure 3-8: Interface with Customer STD

4 External Interface Requirements

4.1 User Interfaces

The front-end, client facing User Interface (UI) elements for this product will be written in JavaScript, which is natively supported in all modern browsers. A focus will be made to ensure that functionality and user experience is maintained across Apple and Android devices, as well as modern desktop browsers (Chrome, Firefox, Safari and Edge).

A/B testing (or split testing) should be completed as part of the product development to test UI elements (layouts, colours, interface structures, etc.). User data generated while browsing the site (e.g. referral source, search criteria, time spent on the site, when the user left the site, etc.) will be analysed to generate new UI configurations, and A/B tested. Machine learning should be applied to generate more insight into the potential site tweaks.

The UI should also follow the guidelines for *User Interface Design Basics*, developed by the US Department of Health and Human Services [2]. The exact requirements for the user interface will be outlined in a separate document. However, as a reference, the client (traveller) UI should follow the format shown in Figure 4-1 below.

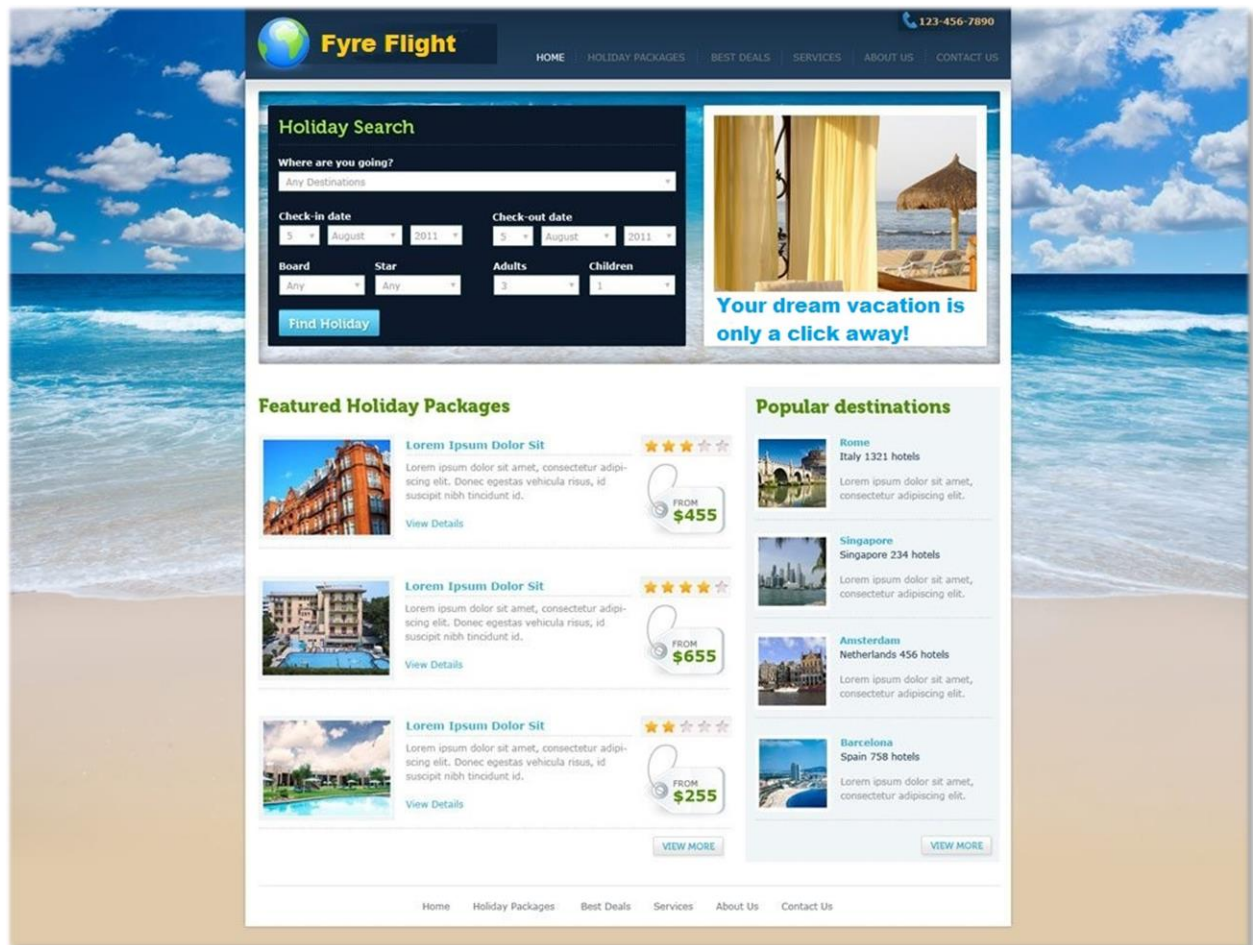


Figure 4-1: Client UI Template

Original source: <https://www.phpjabbbers.com/free-travel-website-template-141.php>

4.2 Hardware Interface

Servers / hardware will be implemented to:

- Implement client facing frontend elements (Web App Servers)
- Store site usage data, client booking and payment details, client account details. (Data Warehouses)
- Servers to support backend functions and implement vendor facing API elements (Job Servers)
- Implement backend elements

Each of these elements will be at least one server instance, which will physically be located on an Amazon Web Service (AWS) server array.

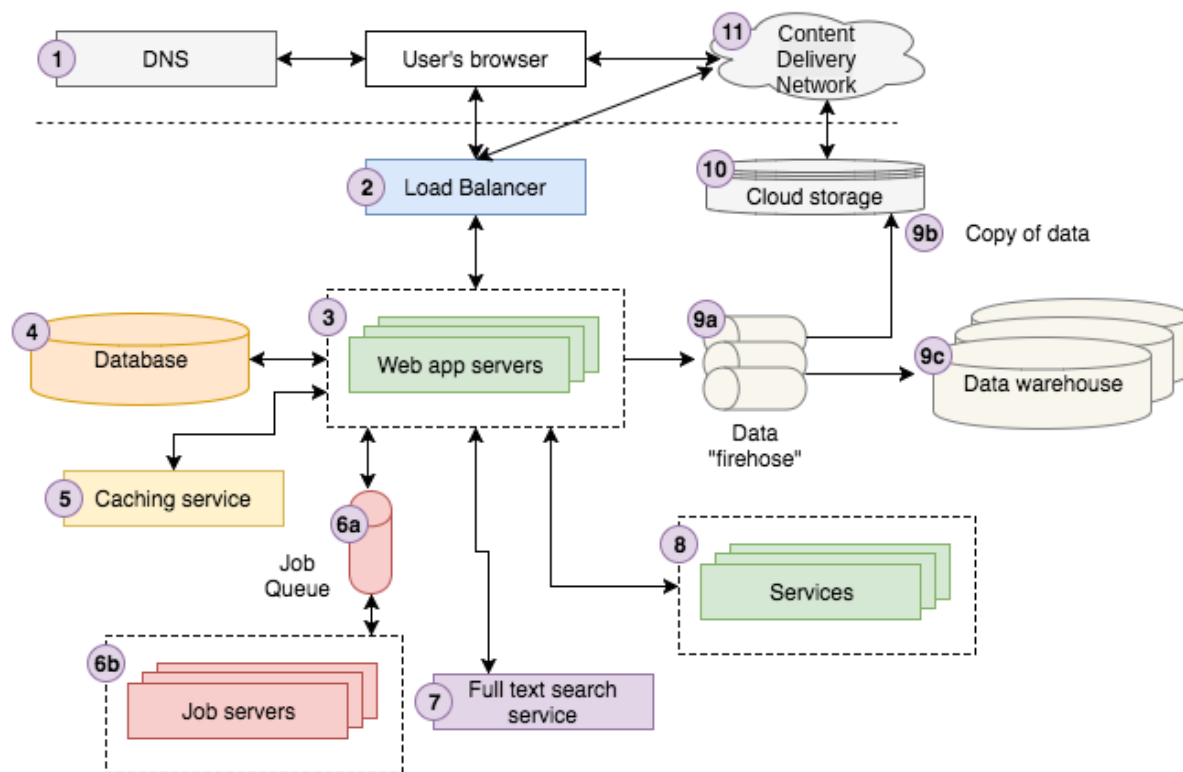


Figure 4-2: Hardware Interface

Original source: <https://www.phpjabbbers.com/free-travel-website-template-141.php>

4.3 Software Interfaces

The client facing software will render elements into html using JavaScript for display on a modern browser. The backend will be written in python, and the interfaces with the vendor APIs will be configured based on their standards.

4.4 Communications Interfaces

Initial releases will communicate with the client via their web browser on the website front end, and via email. All traffic between frontend and backend will be encrypted with https at a minimum, and all payment information will meet web e-commerce standards.

Future integration with social media is possible.

5 Non-functional Requirements/ Quality Requirements

5.1 Performance Requirements

The system should be fast for the user to be able to search and book travel. The system should be visibly appealing so that the users should want to stay and feel comfortable doing research on the site. The site should be able to maintain speed to support multiple users booking travel at one time.

The site should contain a Google Analytics tag provided by the client so that they can maintain data on the site's users and be able to better advertise and remarket to their users. The backend of the site should be user friendly for the client to make copy changes easily.

5.2 Security Requirements

This software has 2 types of software requirements: storing of user data and payment processing.

For user data, our site must be secure and block any types of unauthorized use of a user's account.

For payment processing, our connection to the payment processing software must be airtight. The payment must be quick but provide the user with comfort that their information is secure.

5.3 Portability Requirements

The site should work on mobile devices, tablets, and desktop computers. It should also work on all type of browsers including Chrome, Firefox, Safari and Edge.

6 Prioritization and Release Plan

6.1 Product development prioritization

Product development and release should follow a rapid release schedule. The goal for product release will be to have the unique value proposition [element(s) which differentiate the product from existing competitors] released prior to the completion of the remaining elements. In this way, an iterative approach can be taken towards developing innovative aspects of the platform, while at the same time the remaining base functionality can be filled in along the way.

By pushing high risk / differentiating features ahead, they can be tested in parallel with the development of lower risk aspects. Focusing early on high risk features will allow time to rework the product if they are not well received or do not work as expected.

An initial product prototype release should be made within the first couple weeks of the development process to facilitate UI testing in parallel with backend development.

Later, a backend prototype should be developed to allow UX testing while developing the underlying functionality.

6.2 Development milestones

Below are the project milestones. The focus is to release at the earliest possible time so that we can begin generating data and insights into the product function as early as possible at each step.

- **November 15, 2019 - Frontend Prototype I**
 - Proof of concept. Basic UI functionality, focusing on features that would uniquely differentiate the product from the competition. Data logging should be included to begin generating a data set to inform design choices going forward.
- **December 3, 2019 - Frontend Prototype II**
 - Second iteration of the front-end prototype, with enough functionality to fully simulate the user experience.
- **December 15, 2019- Full Stack Prototype I**
 - Full stack prototype. Back end should have all functionality to simulate the function of all client facing elements.
- **January 1, 2020- Soft public release**
 - All functionality should exist in the product. Users can make use of all site functionality but may be restricted on their ability to book and pay money, depending on team's confidence in the robustness of security and functionality. [weighting the gains of earlier release against consequences of security breach or adverse impact to users] User base may be small enough that technical problems involving paid bookings can be resolved on a case-by-case basis, if the resources exist for this.

- **February 1, 2020- Hard public release**
 - All functionality is implemented, services are live. Promotion is pushed hard to attempt to build market traction. User metrics are recorded and tracked closely. Features and UI changes are A/B tested for impact on user experience.

6.3 Feature Release List

1. Interface with the vendors
2. Basic user interface
3. Search abilities
4. Payment processing
5. Manage Booking
6. Administrative
7. Improved user interface
8. Drive Sales

6.4 Feature release plan

Feature release list section is mapped into the release plan table below.

Feature ID	Priority Level	Type of Feature	Release Date	Remarks
FR 2	1	F	Nov 15, 2019	Front End UI Proof of concept.
FR 1-A	2	F	Dec 3, 2019	Initially create interface with small number of vendors, prioritize additional vendors based on user input as traction increases.
FR 1-B	1	F	Feb 30, 2020	Interface with larger number of vendors. (FR 1-C not shown, is adding additional vendors past this date)
FR 3	1	F	Dec 15, 2019	Search functions to navigate vendor data.
FR 5	2	F	Dec 15, 2019	Proof of concept for the backend
FR 6-A	2	F	Dec 3, 2019	Basic data logging to generate site usage data for the frontend prototype.
FR 6-B	1	F	Jan 1, 2020	More advanced site usage tracking
FR 4	1	F	Feb 30, 2020	Users can now book and pay for things. Depending on confidence in protecting users against adverse outcomes, this could be released earlier.
FR 7, FR 8	2	F	Feb 30, 2020	Promotion is used to build product traction; UI changes are tested against user metrics.

Appendices

Appendix A: Data Flow Diagrams

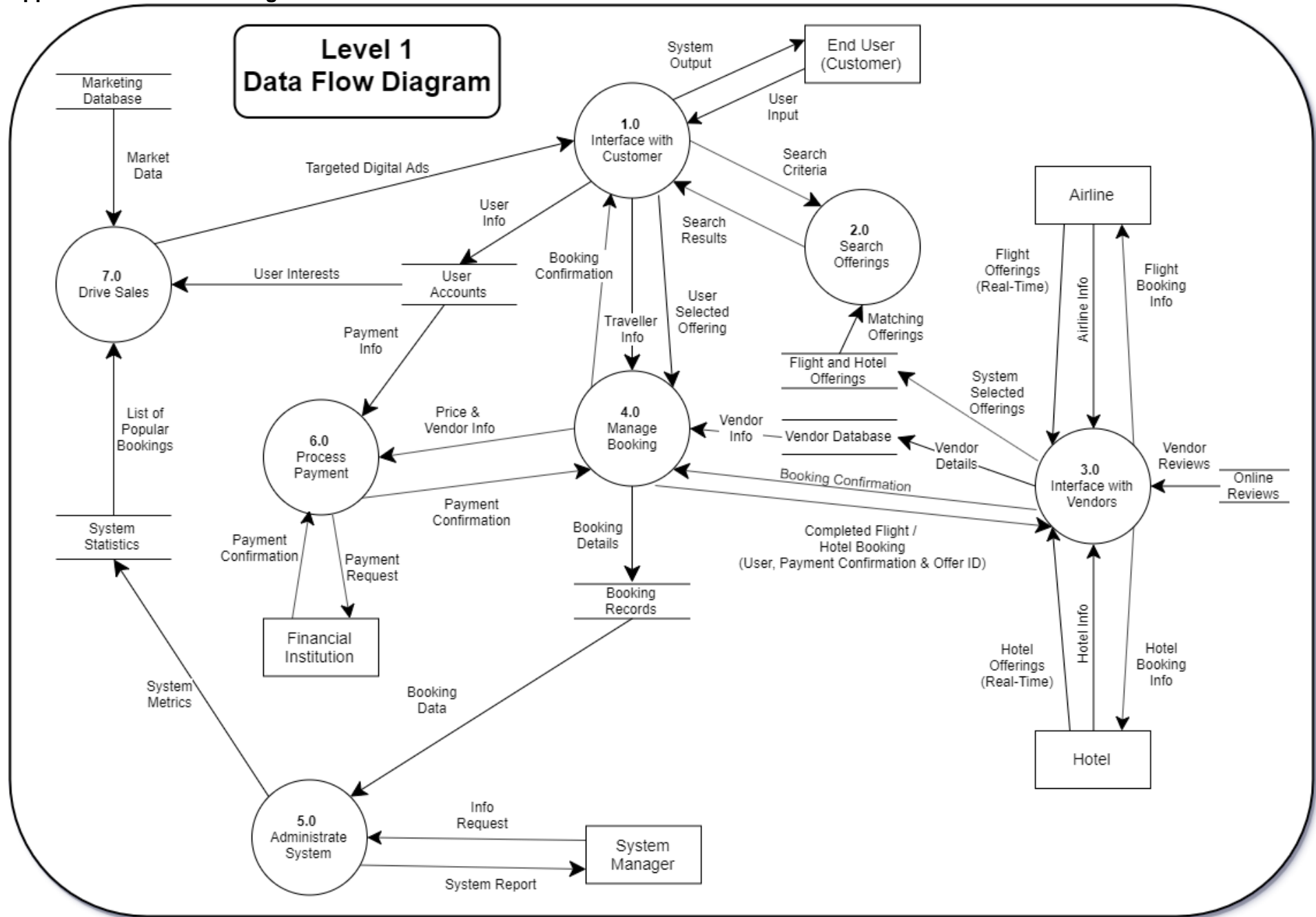
Includes the complete systems process diagrams at the different levels

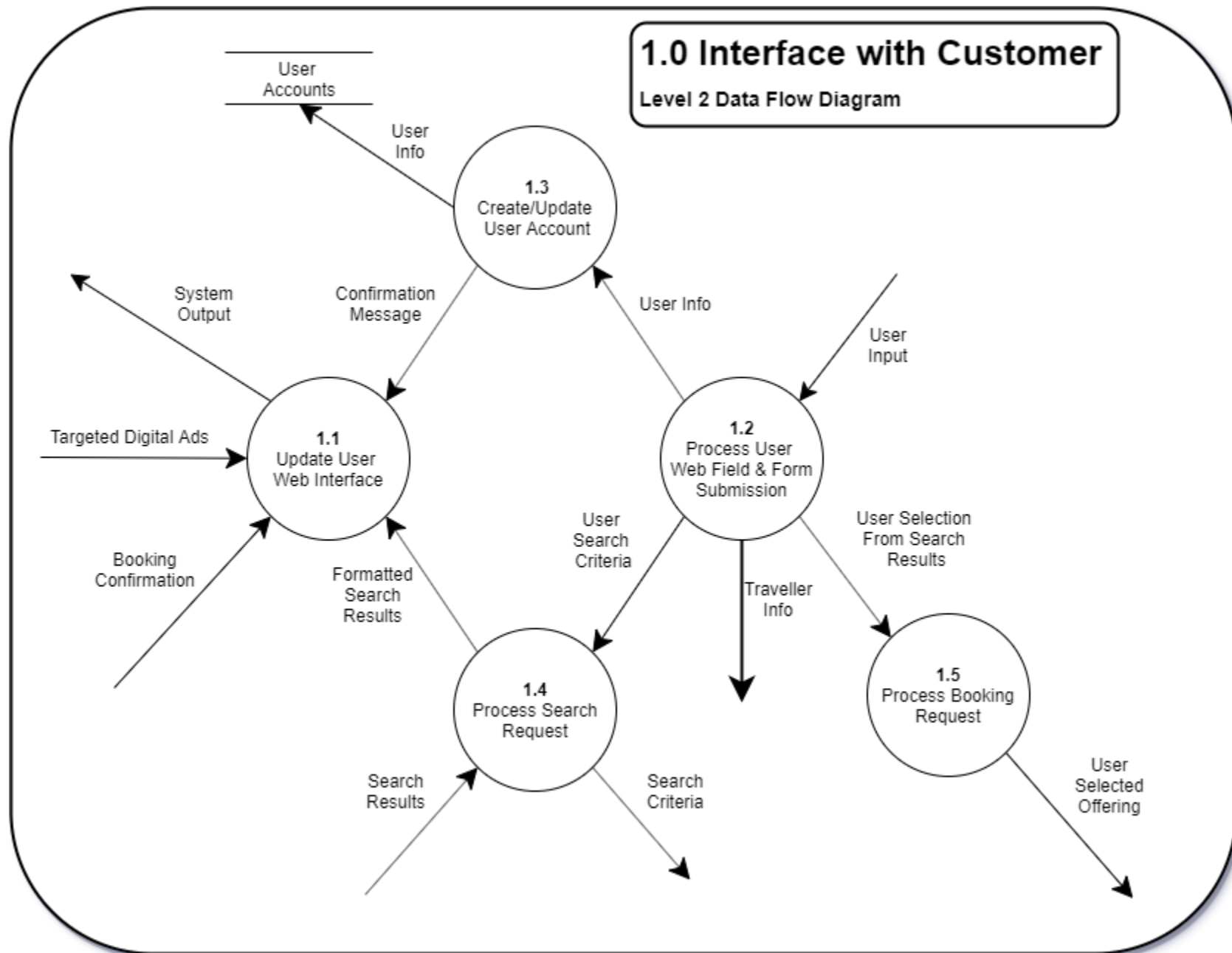
Appendix B: Entity Relationship Diagrams

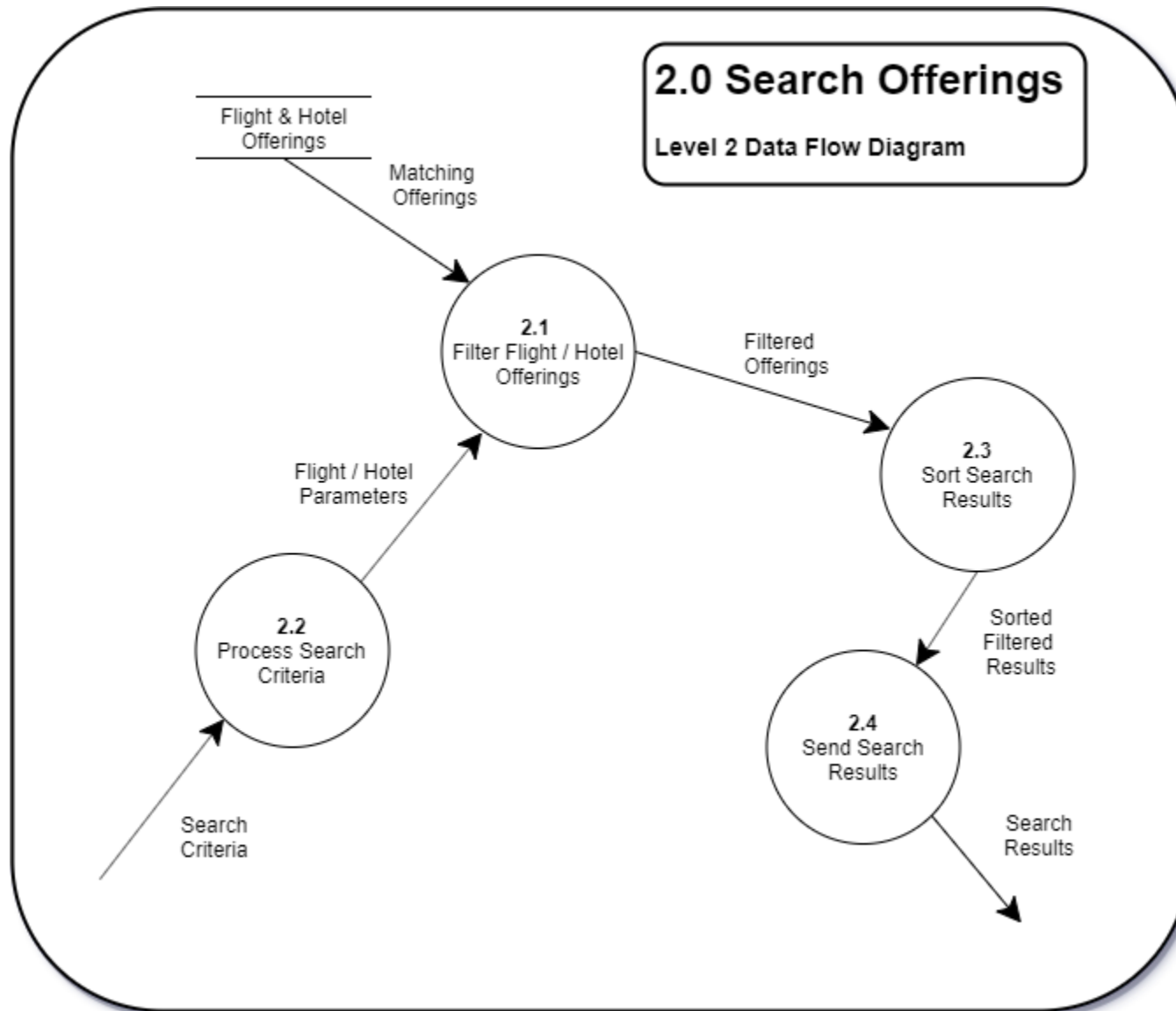
Includes system's Entity Relationship Diagrams

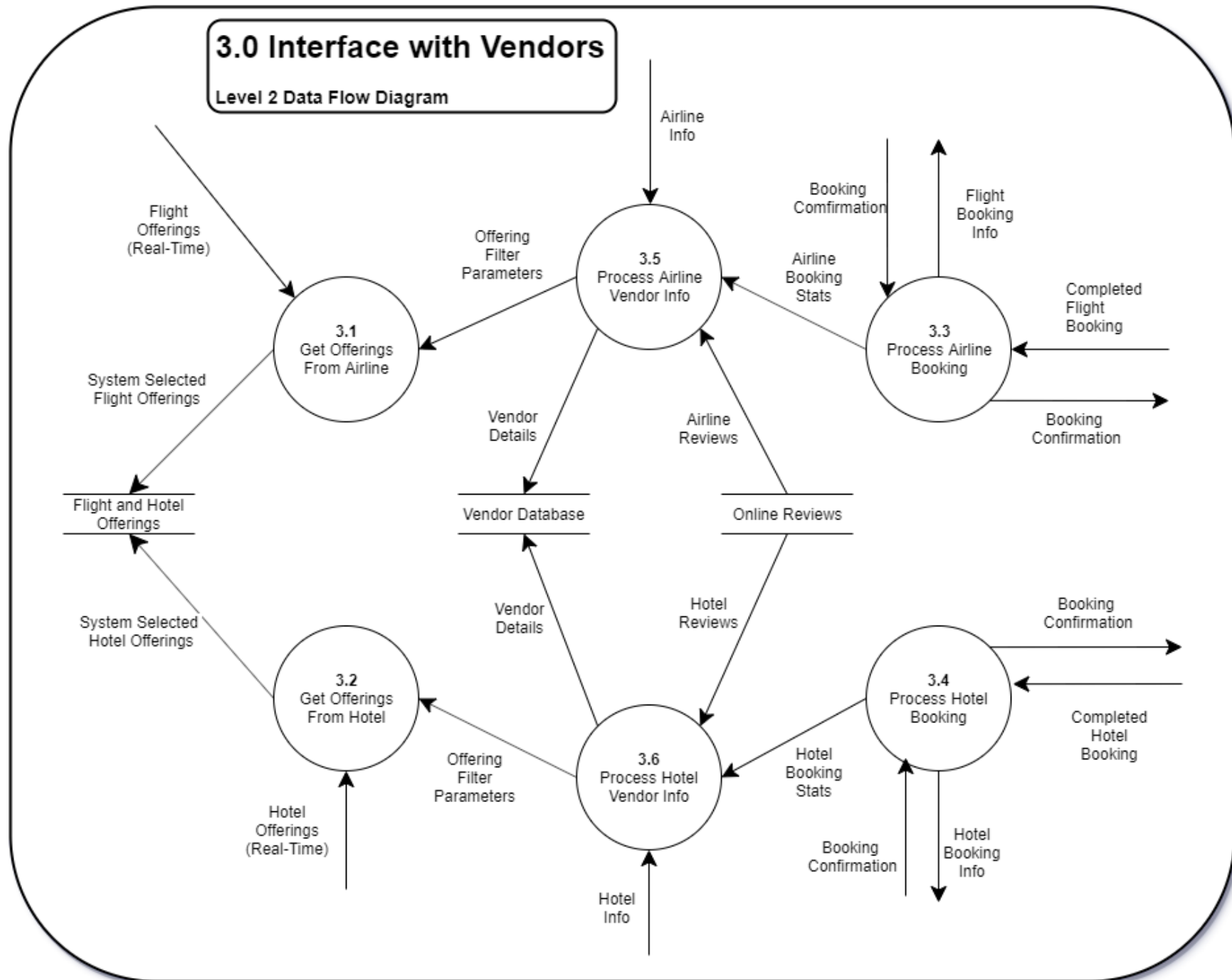
Appendix C: Data Dictionary

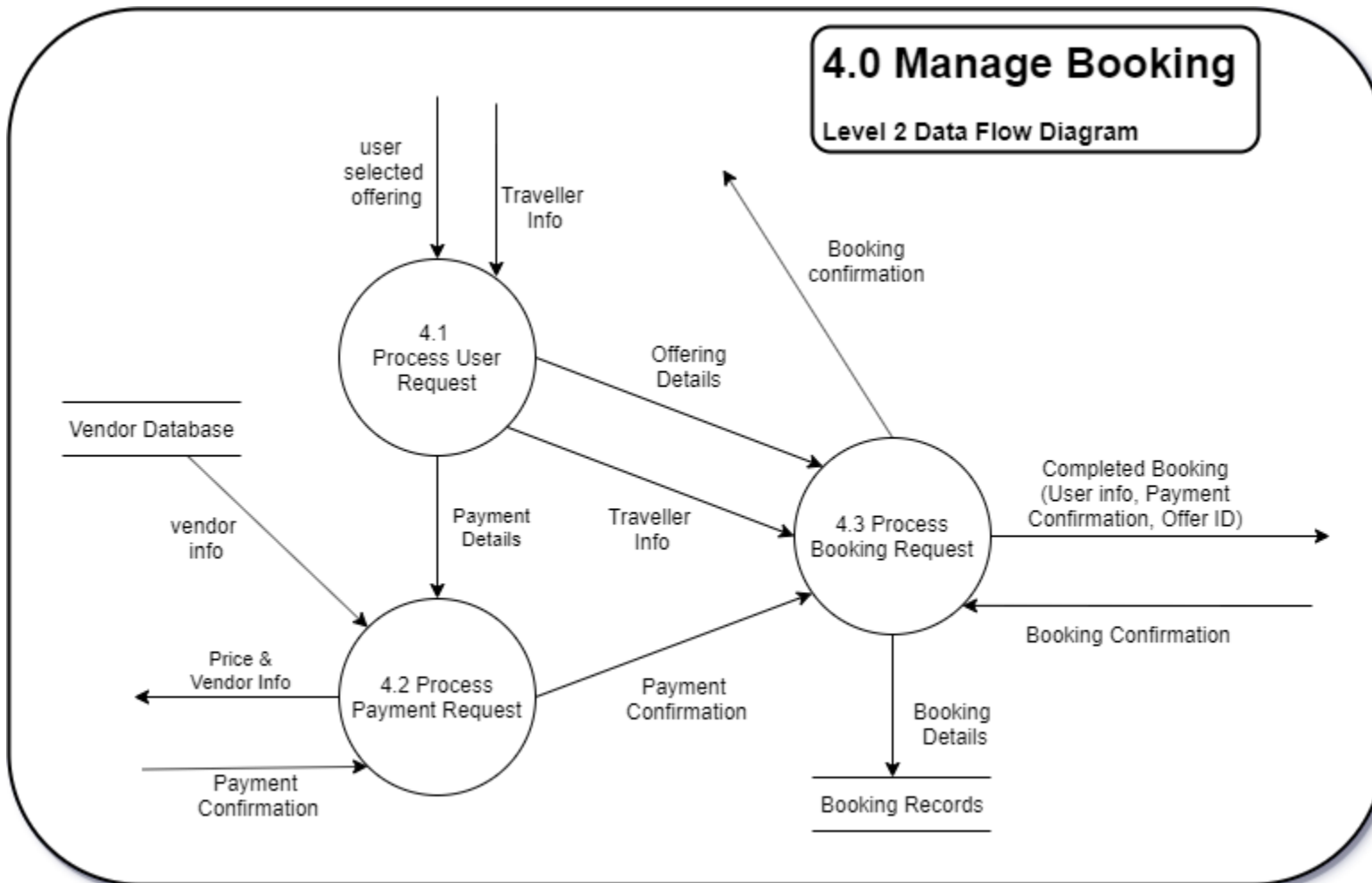
Includes system's Data Dictionary

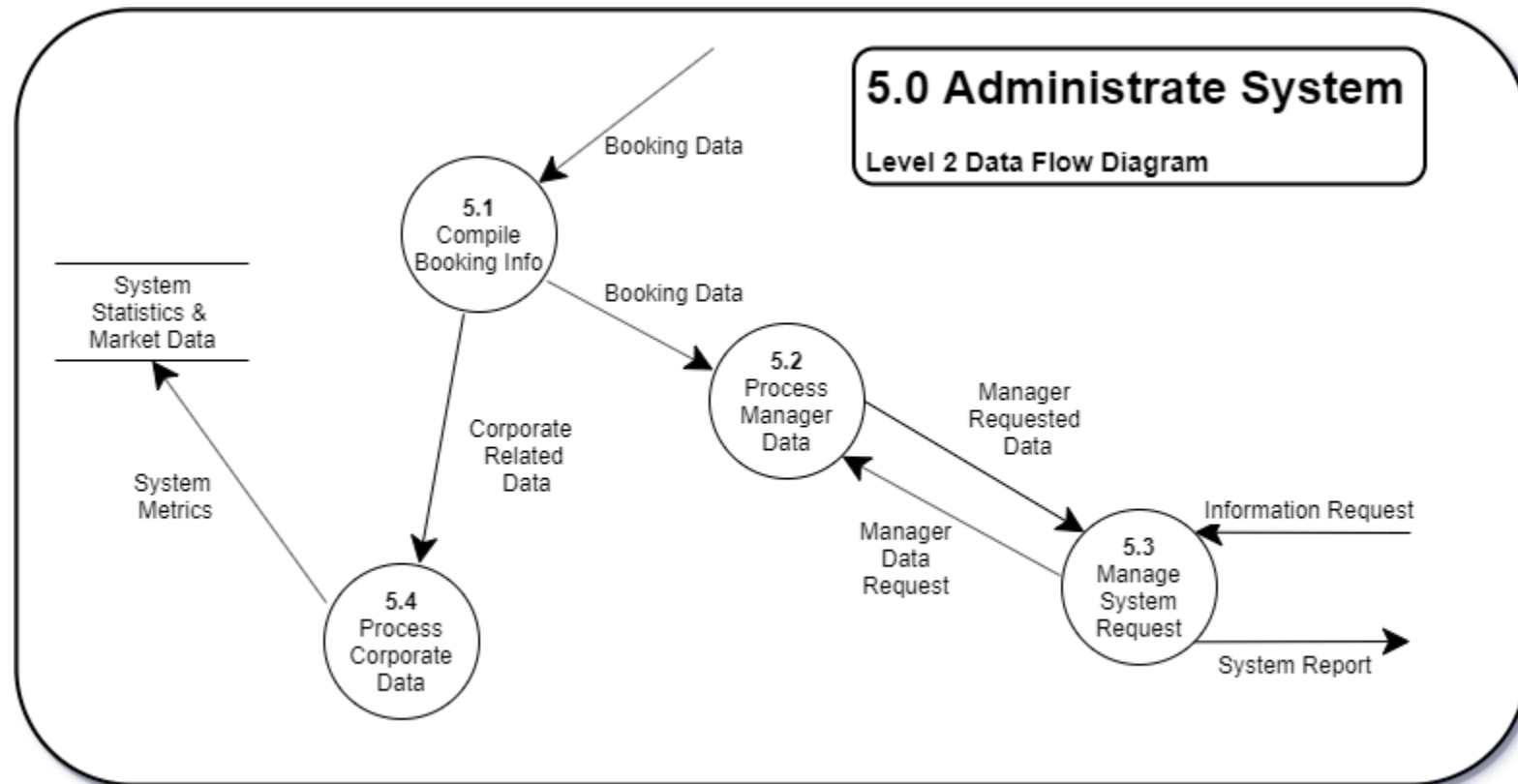


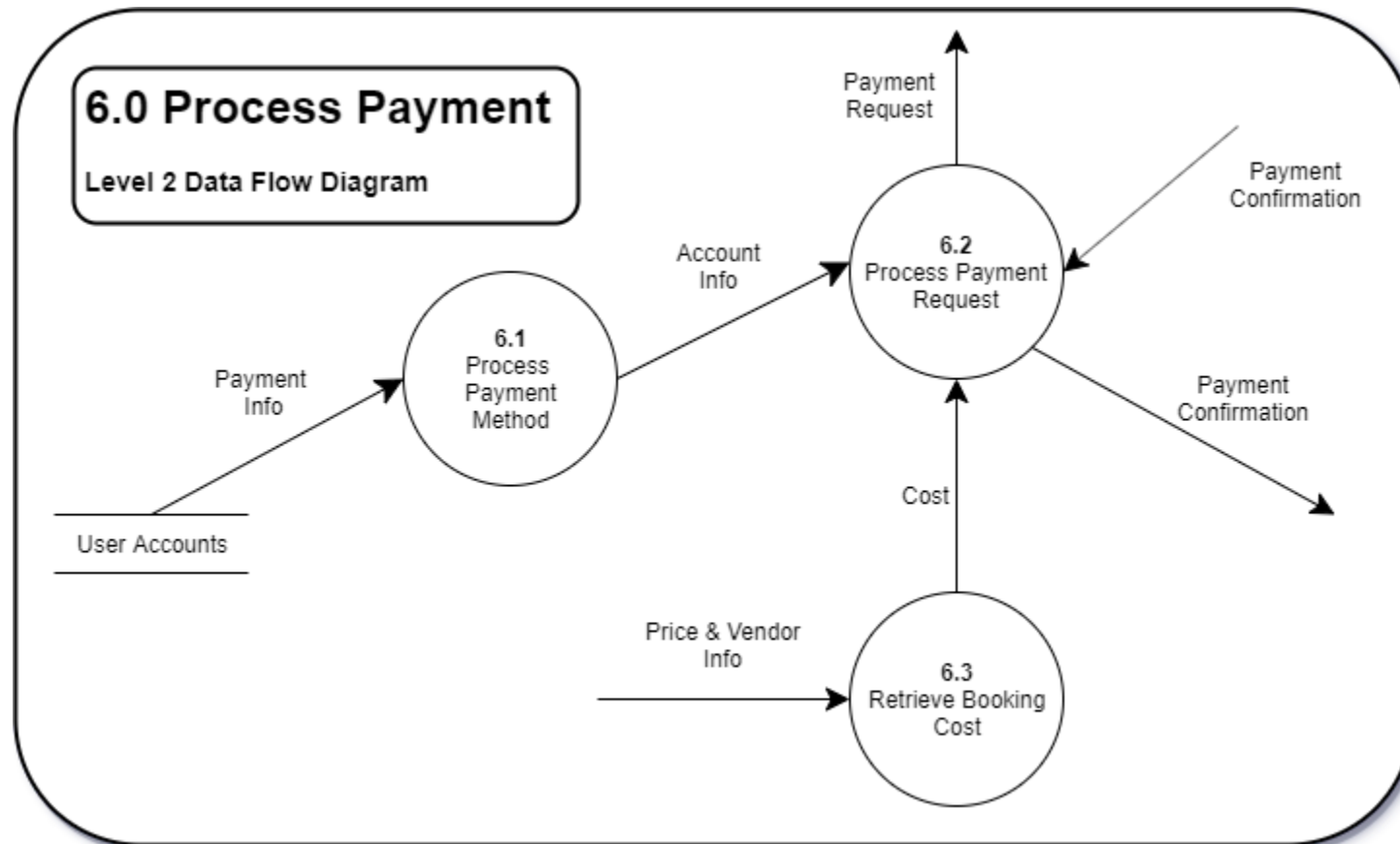


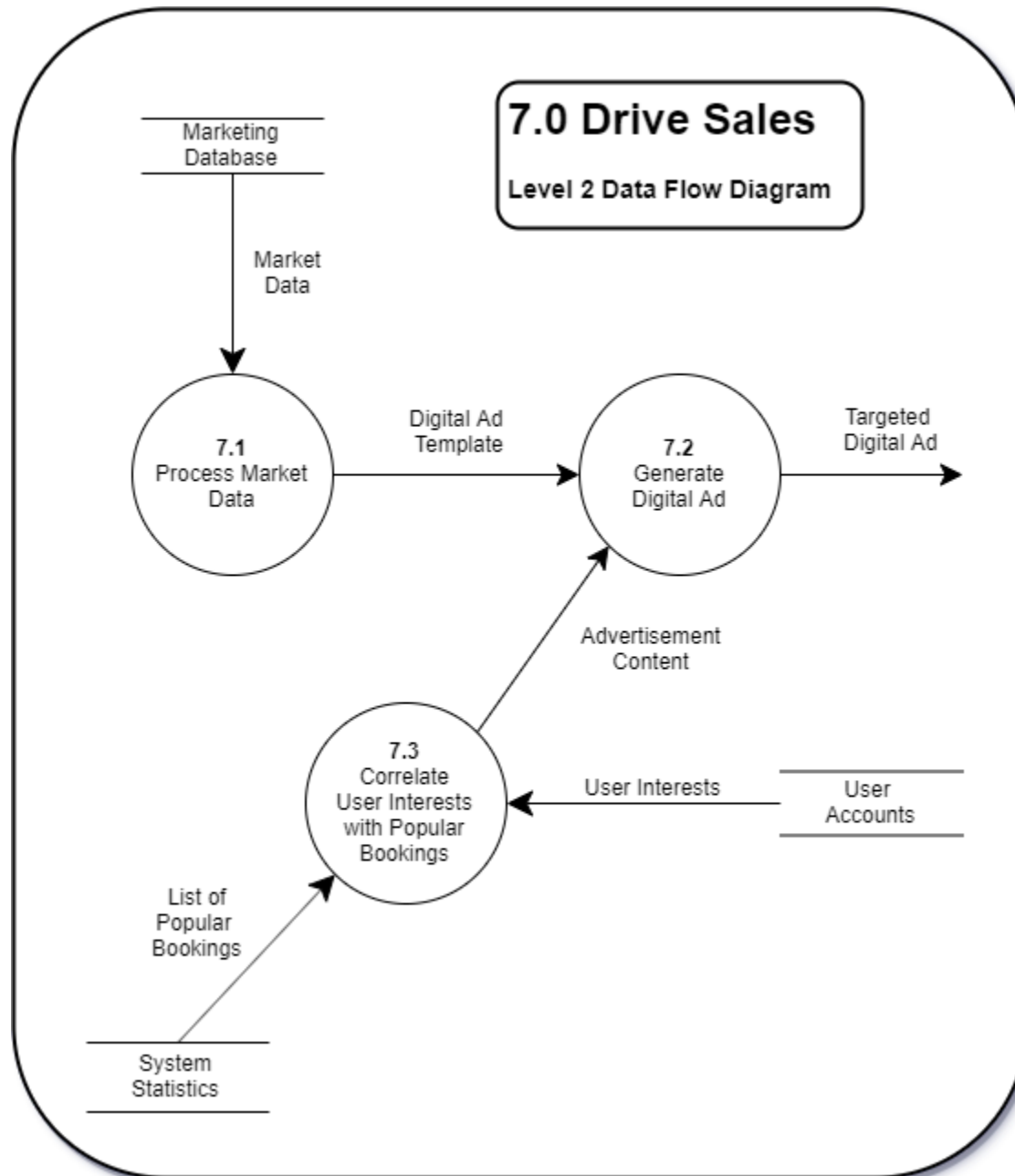




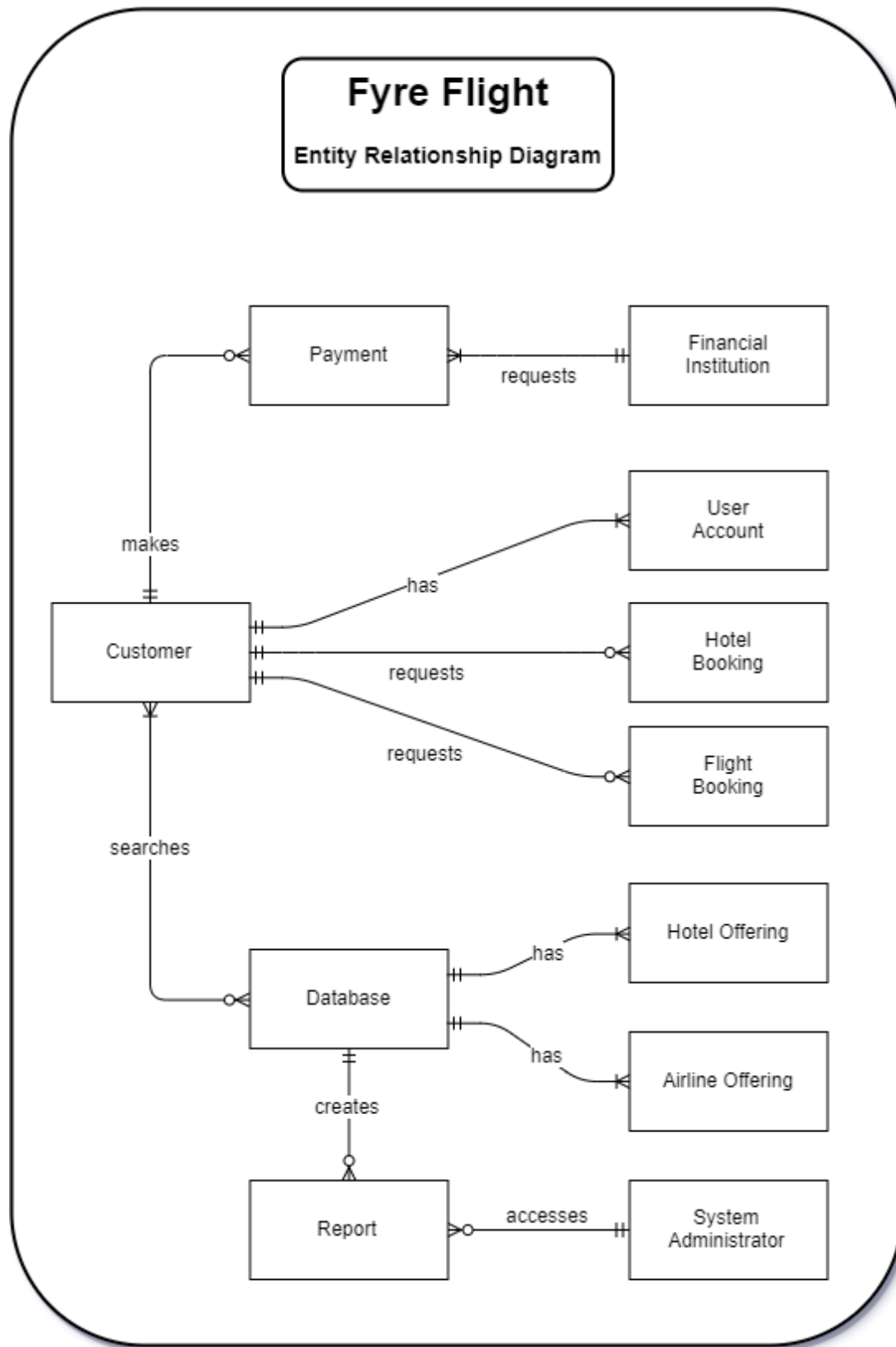








Appendix B: Entity Relationship Diagram



Entity (Class)	Field	Description	Format
Customer	customerID	Unique id number	integer
	firstName	Customer's first name	string
	lastName	Customer's last name	string
	dateOfBirth	Customer's D.O.B	YYYY/MM/DD
Payment	paymentID	Unique id number	8 digit integer
	cardType	Type id	string
	cardNumber	12 digit integer	integer
	expirationDate	4 digit integer	integer
	paymentAmount	Charge to customer	double
Financial Institution	bankName	Corporate Name	string
	institutionNumber	MICR Number of Bank	3 digit integer
	branchNumber	Branch	integer
	accountNumber	Account Number	integer
User Account	customerID	Customer that owns this account	integer
	loginName	Customer's unique login	string
	loginPassword	Customer's login password	string
	bookings	IDs of past bookings	Integer Array
	savedPaymentInfo	Credit card info on file	json
	userInterests	List of vacation interests identified by the user	json
Hotel Booking	bookingID	Unique identifier	8 digit integer
	customerID	ID of Customer who made booking	8 digit integer
	hotelOfferID	Unique identifier	8 digit integer
	arrivalDate	Checkin Date	YYYY/MM/DD
	departureDate	Checkout Date	YYYY/MM/DD
	totalPrice	CAD including tax	double
Flight Booking	bookingID	Flight unique identifier	8 digit integer
	customerID	Unique identifier	8 digit integer
	flightOfferID	Assigned ID	8 digit integer
Hotel Offering	hotelOfferID	Assigned ID	8 digit integer
	hotelName	Name of hotel	string
	hotelAddress	Address of hotel	string
	roomType	Bed type (king, queen, double) and room type (suite, junior, etc.)	string
	ratePerNight	Cost of hotel per night	double
	firstDateAvailable	First possible checkin date	YYYY/MM/DD
	lastDateAvailable	Last possible checkout date	YYYY/MM/DD
	flightOfferID	Unique identifier	8 digit integer
Flight Offering	airline	Identifier for airline	string
	flightNumber	Identifier for route	string
	departureAirportCode	Identifier for departure airport	3 character string
	arrivalAirportCode	Identifier for arrival airport	3 character string
	departureDate	Date of leaving origin	YYYY/MM/DD
	arrivalDate	Date of Leaving Destination	YYYY/MM/DD
	totalPrice	Total Price of Flight	double

Entity (Class)	Field	Description	Format
System Administrator	adminLoginID	Username of User	string
	adminPassword	Unique Password	string
	reports	Titles of Reports	Array of strings
Report	customerIDs	Unique identifier	8 digit integer
	bookingIDs	Set of ID numbers	json
	reportPeriodStartDate	Report Start Date	YYYY/MM/DD
	reportPeriodEndDate	Report End Date	YYYY/MM/DD
	reportPeriodTotalRevenue	CAD dollars before tax	double
Database	customers	List of customer accounts	Array of pointers
	bookings	List of bookings	Array of pointers
	hotelOffers	List of hotel offerings	Array of pointers
	airlineOffers	List of airline offerings	Array of pointers
	payments	List of payment details	Array of pointers