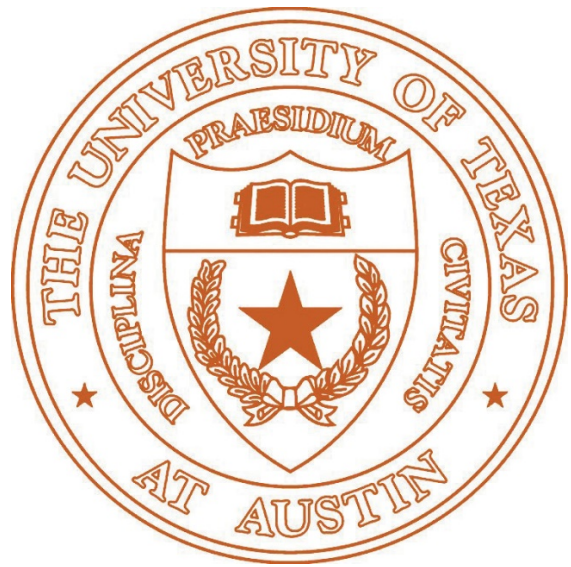


**University of Texas at Austin, Cockrell School of Engineering**  
**Software Architecture – EE 382C.7**



**Assignment # 2**  
**Derivation and Evaluation of Business Blueprints**  
April 01, 2016

Gabrielson Eapen  
EID: EAPENGP

## Table of Contents

<b>2. Derivation and Evaluation of Business Blueprints .....</b>	<b>3</b>
<b>2.1 Prioritization of Stakeholder Qualities/Constraints along with Quality Categories.....</b>	<b>3</b>
<b>2.2 Business Blueprint Derivation.....</b>	<b>5</b>
<b>2.2.1 Graphical Representation of Components and their Relations.....</b>	<b>5</b>
<b>2.2.2 Textual Representation of Components and their Relations .....</b>	<b>6</b>
<b>2.2.2.1 Allocation of functions and data to components .....</b>	<b>6</b>
<b>2.2.2.2 Function I/O Dependencies between Components.....</b>	<b>7</b>
<b>2.2.2.3 Function I/O Dependencies between Components and External Producers/Consumers</b>	<b>8</b>
<b>2.2.2.4 Function I/O Satisfied within the Same Component .....</b>	<b>9</b>
<b>2.3 Derivation Plan and Rationale .....</b>	<b>11</b>
<b>2.3.1 Derivation Plan.....</b>	<b>11</b>
<b>2.3.2 Potential Conflicts and Impact on Derivation Plan .....</b>	<b>12</b>
<b>2.4 Evaluate Business Blueprint Structure .....</b>	<b>12</b>
<b>2.4.1 Coupling and Cohesion Metrics .....</b>	<b>12</b>
<b>2.4.1.1 Number of Inputs/Outputs between components.....</b>	<b>12</b>
<b>2.4.1.2 Number of dependencies between components.....</b>	<b>13</b>
<b>2.4.1.3 Degree of Cohesion .....</b>	<b>13</b>
<b>2.4.2 Size and Complexity Metrics.....</b>	<b>13</b>
<b>2.4.2.1 Number of functions allocated to a component.....</b>	<b>13</b>
<b>2.4.2.2 Number of data elements allocated to a component .....</b>	<b>13</b>
<b>2.4.2.3 Number of components in the blueprint .....</b>	<b>14</b>
<b>2.4.2.4 Component Complexity .....</b>	<b>14</b>
<b>2.4.3 Support for Applied Heuristic .....</b>	<b>14</b>

## 2. Derivation and Evaluation of Business Blueprints

### 2.1 Prioritization of Stakeholder Qualities/Constraints along with Quality Categories

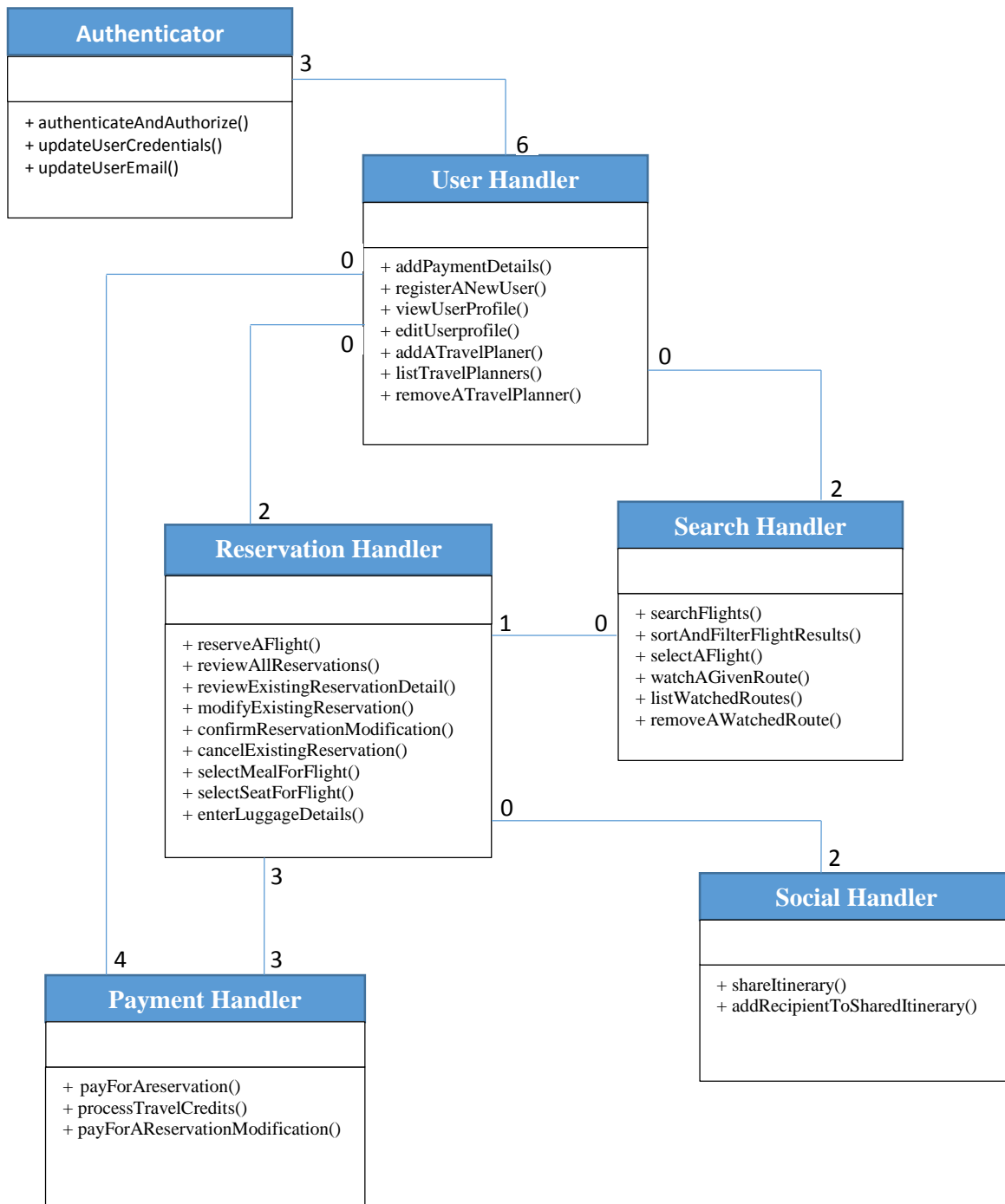
The following is a prioritized table of Stakeholder Qualities and Constraints presented in the problem domain.

Priority	Need/Quality	Classification	Priority Justification
1	The site (application) should be available to end-users for the entire duration of their access (visit). Specifically, the system needs to be available 24x7 with a four 9's (99.99%) uptime goal.	Availability	If travelers are unable to access the site (application) when they need it, they are unable to schedule and plan trips. Besides inconveniencing the traveler, it adversely affects the reputation of the site.
2	The application user interface (UI) should be simplistic and intuitive to use so that end-user needs minimal to no training in order to navigate the site.	Usability	If travelers find it difficult to use or navigate the site, they are less likely to use it. Ease of use leads to higher end-user acceptance and satisfaction.
3	The user should not have to wait for extended periods of time at any point when using the system.	Performance	Since the application is available globally 24x7, end-user usage patterns can vary. But regardless of application load, all application functions must perform consistently and complete in the specified amount of time.
4	Authenticated access to the application should be secure and user credentials should not be compromised. SSL can be used to secure the sensitive information. Sensitive personal or payment information should be encrypted when stored in the database.	Security	The application must comply with industry standards in storage and transmission of sensitive personal information to avoid any legal liability. Specifically collection, transmission, and storage of payment (CREDIT CARD) information or cardholder data must comply with the Payment Card Industry Data Security Standard (PCI DSS).
5	The application should be able to handle large numbers of users and have ability to increase capacity on demand in response to growth or during peak travel seasons.	Scalability	To ensure consistent performance, when user load or activity increased, the application infrastructure needs to be elastic and scale to meet the higher load. This helps satisfy priorities 1 and 3 as well.

6	The application should be easily extensible	Extensibility	As our company desires to agile and aggressive in the market place, it needs to be able to add new features or capabilities with minimal design effort. So tight coupling between modules should be avoided.
7	There should be regularly scheduled backups of critical user and system data to allow for recovery whenever needed.	Data backup and recovery	Hardware failures happen and it would be embarrassing to inform customers that we have lost their data. If possible, continuous or almost real-time backups should also exist to allow restores to a specific point-in-time.
8	The application should be created with proper coding principles and design patterns with controlled releases from development to test to production.	Maintainability	Similar to priority 6, incremental updates and/or bug fixes need to have controlled rollouts (or rollbacks) without availability disruption.
9	Delivery of the application must occur within the agreed upon time frames.	Project Schedule	Our company desires to be agile and aggressive in the market place. It is important to remain on the proposed release schedule for the application to be successful.
10	The application should be created within the specified budget without cost overrun.	Project Cost	Adhering to the stipulated budget goes a long way in inspiring confidence in our company's long term viability.

## 2.2 Business Blueprint Derivation

### 2.2.1 Graphical Representation of Components and their Relations



## 2.2.2 Textual Representation of Components and their Relations

From the given domain, a set of components has been extracted. The section is dedicated to present a textual representation of the mapping between such components with data and functions, as well as the relations and dependencies between components.

### 2.2.2.1 Allocation of functions and data to components

Component	Data	Functions
Authenticator	User Credentials User Email	authenticateAndAuthorize() updateUserCredentials() updateUserEmail()
Payment Handler	Credit Amount	payForAReservation() processTravelCredits() payForAReservationModification()
Reservation Handler	User Reservation List Trip Reservation Details <sup>1</sup>	reserveAFlight() reviewAllReservations() reviewExistingReservationDetail() modifyExistingReservation() confirmReservationModification() cancelExistingReservation() selectMealForFlight() selectSeatForFlight() enterLuggageDetails()
User Handler	User ID User Profile Information <sup>2</sup> Planner Information <sup>3</sup> Payment Information	addPaymentDetails() registerANewUser() viewUserProfile() editUserprofile() addATravelPlanner() listTravelPlanners() removeATravelPlanner()
Search Handler	Watched Routes	searchFlights() sortAndFilterFlightResults() selectAFlight() watchAGivenRoute() listWatchedRoutes() removeAWatchedRoute()

---

<sup>1</sup> Trip Reservation details has all the information in the reservation record including trip cost and shared recipients

<sup>2</sup> Profile Information is specified as a single data entity but excludes Email Address

<sup>3</sup> Planner information is specified as a single data entity and includes name and email

Social Handler	Shared Reservation List Reservation Shared Recipients	shareItinerary() addRecipientToSharedItinerary()
----------------	--	---

#### 2.2.2.2 Function I/O Dependencies between Components

Components	Dependency
FROM: Authenticator TO: User Handler	<ul style="list-style-type: none"> <li>“addPaymentDetails” requires “authenticated user id” from “authenticateAndAuthorize”</li> <li>“viewUserProfile” requires “authenticated user id” from “authenticateAndAuthorize”</li> <li>“editUserProfile” requires “authenticated user id” from “authenticateAndAuthorize”</li> <li>“addATravelPlanner” requires “authenticated user id” from “authenticateAndAuthorize”</li> <li>“listTravelPlanners” requires “authenticated user id” from “authenticateAndAuthorize”</li> <li>“removeATravelPlanner” requires “authenticated user id” from “authenticateAndAuthorize”</li> </ul>
FROM: User Handler TO: Authenticator	<ul style="list-style-type: none"> <li>“updateUserCredentials” requires user submitted “username” from “registerANewUser”</li> <li>“updateUserCredentials” requires user submitted “password” from “registerANewUser”</li> <li>“updateUserEmail” requires user submitted “email address” from “registerANewUser”</li> </ul>
FROM: User Handler TO: Reservation Handler	<ul style="list-style-type: none"> <li>“reserveAFlight” requires data “User ID” which was allocated to component “User Handler”</li> <li>“reviewAllReservations” requires data “User ID” which was allocated to component “User Handler”</li> </ul>
FROM: Reservation Handler TO: Social Handler	<ul style="list-style-type: none"> <li>“shareItinerary” requires data “Trip Reservation Details” which was allocated to component “Reservation Handler”</li> <li>“addRecipientToSharedItinerary” requires data “Trip Reservation Details” which was allocated to component “Reservation Handler”</li> </ul>
FROM: User Handler TO: Search Handler	<ul style="list-style-type: none"> <li>“watchAGivenRoute” requires data “User ID” which was allocated to component “User Handler”</li> <li>“listAGivenRoute” requires data “User ID” which was allocated to component “User Handler”</li> <li>“removeAGivenRoute” requires data “User ID” which was allocated to component “User Handler”</li> </ul>

FROM: Search Handler TO: Reservation Handler	<ul style="list-style-type: none"> <li>• “reserveAFlight” requires “flight details” from “selectAFlight”</li> </ul>
FROM: Payment Handler TO: Reservation Handler	<ul style="list-style-type: none"> <li>• “reserveAFlight” requires “paymentSuccess” from “payForAReservation”</li> <li>• “confirmReservationModification” requires “paymentSuccess” from “payForAReservationModification”</li> <li>• “cancelExistingReservation” requires data “Credit Amount” which was allocated to component “Payment Handler”</li> </ul>
FROM: Reservation Handler TO: Payment Handler	<ul style="list-style-type: none"> <li>• “payForAReservation” requires data “Trip Reservation Details” which was allocated to component “Reservation Handler”</li> <li>• “payForAReservationModification” requires data “Trip Reservation Details” which was allocated to component “Reservation Handler”</li> <li>• “processTravelCredits” requires data “Trip Reservation Details” which was allocated to component “Reservation Handler”</li> </ul>
FROM: User Handler TO: Payment Handler	<ul style="list-style-type: none"> <li>• “payForAReservation” requires data “Payment Information” which were allocated to component “User Handler”</li> <li>• “payForAReservation” requires data “User ID” which were allocated to component “User Handler”</li> <li>• “payForAReservationModification” requires data “Payment Information” which were allocated to component “User Handler”</li> <li>• “payForAReservationModification” requires data “User ID” which were allocated to component “User Handler”</li> </ul>

### 2.2.2.3 Function I/O Dependencies between Components and External Producers/Consumers

Components	Dependency
TO: Authenticator	<ul style="list-style-type: none"> <li>• “authenticateAndAuthorize” requires “Current or New Session ID” from external</li> </ul>
FROM: Authenticator	<ul style="list-style-type: none"> <li>• “authenticateAndAuthorize” sends “Session ID Update Timestamp” from external</li> </ul>
TO: Search Handler	<ul style="list-style-type: none"> <li>• “searchFlights” requires “Airport List”, “All Flights Schedules”, and Calendar database from external</li> </ul>
FROM: Search Handler	<ul style="list-style-type: none"> <li>• “watchAGivenRoute” sends “Watched Routes Addition” to external</li> <li>• “removeAGivenRoute” sends “Watched Routes Removal” to external</li> </ul>



TO: Reservation Handler	<ul style="list-style-type: none"> <li>• “reserveAFlight” requires “Passenger Name Record (PNR)” from external</li> </ul>
FROM: Reservation Handler	<ul style="list-style-type: none"> <li>• “reserveAFlight” sends “Passenger Information” and “Itinerary details” for ticketing to external</li> <li>• “confirmReservationModification” sends “Reservation Change Details” to external</li> <li>• “cancelExistingReservation” sends “Reservation Code” to external</li> <li>• “selectMealForFlight” sends “Meal Preferences” to external</li> <li>• “selectSeatForFlight” sends “Seat Choices” to external</li> <li>• “enterLuggageDetails” sends “Baggage Details” to external</li> </ul>
TO: Payment Handler	<ul style="list-style-type: none"> <li>• “payForAReservation” requires “Authorization Code” from external</li> <li>• “payForAReservationModification” requires “Authorization Code” from external</li> <li>• “processTravelCredits” requires “Confirmed Refund Amount” from external</li> </ul>
FROM: Payment Handler	<ul style="list-style-type: none"> <li>• “payForAReservation” sends “Payment Authorization Request” to external</li> <li>• “payForAReservationModification” sends “Payment Authorization Request” to external</li> </ul>

#### 2.2.2.4 Function I/O Satisfied within the Same Component

Components	Dependency
WITHIN: Authenticator	<ul style="list-style-type: none"> <li>• “authenticateAndAuthorize” references data “User Credentials” allocated to “Authenticator”</li> <li>• “updateUserCredentials” references data “User Credentials” allocated to “Authenticator”</li> <li>• “updateUserEmail” references data “User Email” allocated to “Authenticator”</li> </ul>
WITHIN: Payment Handler	<ul style="list-style-type: none"> <li>• “processTravelCredits” references data “Credit Amount” allocated to “Authenticator”</li> </ul>
WITHIN: Reservation Handler	<ul style="list-style-type: none"> <li>• “reserveAFlight” references data “User reservation List” allocated to “Reservation Handler”</li> <li>• “reserveAFlight” references data “Trip Reservation Details” allocated to “Reservation Handler”</li> </ul>

	<ul style="list-style-type: none"> <li>• “reviewAllReservations” references data “User reservation List” allocated to “Reservation Handler”</li> <li>• “reviewExistingReservationDetail” references data “Trip Reservation Details” allocated to “Reservation Handler”</li> <li>• “modifyExistingReservation” references data “Trip Reservation Details” allocated to “Reservation Handler”</li> <li>• “cancelExistingReservation” references data “Trip Reservation Details” allocated to “Reservation Handler”</li> <li>• “selectMealForFlight” references data “Trip Reservation Details” allocated to “Reservation Handler”</li> <li>• “selectSeatForFlight” references data “Trip Reservation Details” allocated to “Reservation Handler”</li> <li>• “enterLuggageDetails” references data “Trip Reservation Details” allocated to “Reservation Handler”</li> </ul>
WITHIN: User Handler	<ul style="list-style-type: none"> <li>• “addPaymentDetails” references data “User ID” allocated to “User Handler”</li> <li>• “viewUserProfile” references data “User ID” allocated to “User Handler”</li> <li>• “editUserProfile” references data “User ID” allocated to “User Handler”</li> <li>• “addATravelPlanner” references data “User ID” allocated to “User Handler”</li> <li>• “listTravelPlanners” references data “User ID” allocated to “User Handler”</li> <li>• “removeATravelPlanner” references data “User ID” allocated to “User Handler”</li> <li>• “addPaymentDetails” references data “Payment Information” allocated to “User Handler”</li> <li>• “registerANewUser” references data “User Profile Information” allocated to “User Handler”</li> <li>• “viewUserProfile” references data “User Profile Information” allocated to “User Handler”</li> <li>• “editUserProfile” references data “User Profile Information” allocated to “User Handler”</li> <li>• “addATravelPlanner” references data “Planner Information” allocated to “User Handler”</li> <li>• “addATravelPlanner” references data “Planner Information” allocated to “User Handler”</li> <li>• “listTravelPlanners” references data “Planner Information” allocated to “User Handler”</li> <li>• “removeATravelPlanner” references data “Planner Information” allocated to “User Handler”</li> </ul>
WITHIN: Search Handler	<ul style="list-style-type: none"> <li>• “watchAGivenRoute” references data “Watched Routes” allocated to “Search Handler”</li> </ul>

	<ul style="list-style-type: none"> <li>• “listWatchedRoutes” references data “Watched Routes” allocated to “Search Handler”</li> <li>• “removeAGivenRoute” references data “Watched Routes” allocated to “Search Handler”</li> </ul>
WITHIN: Social Handler	<ul style="list-style-type: none"> <li>• “shareItinerary” references data “Shared Reservation List” allocated to “Social Handler”</li> <li>• “shareItinerary” references data “Reservation Shared Recipients” allocated to “Social Handler”</li> <li>• “addRecipientToSharedItinerary” references data “Reservation Shared Recipients” allocated to “Social Handler”</li> </ul>

## 2.3 Derivation Plan and Rationale

### 2.3.1 Derivation Plan

The following table represents all prioritized constraints and the heuristic(s) that can be implemented to achieve needs of the stakeholders. Note that greyed-out heuristics have been eliminated from the derivation as they have been found in direct conflict to a heuristic strategy with a higher priority (for more, see point 2.3.2). Also note that priorities “X” and “Y” have been omitted from this table. The reason for that being that implementation of the components has little to do with achieving these constraints.

Priority	Heuristic	Reason	Priority Justification
<b>1. Availability</b>			
<b>2. Usability</b>			
<b>3. Performance</b>			
<b>4. Security</b>			
<b>5. Scalability</b>			

<b>6. Extensibility</b>			
<b>7. Data backup and recovery</b>			
<b>8. Maintainability</b>			
<b>9. Project Schedule</b>			
<b>10. Project Cost</b>			

### 2.3.2 Potential Conflicts and Impact on Derivation Plan

Heuristic	Potential Conflict	Possible Relocation

## 2.4 Evaluate Business Blueprint Structure

### 2.4.1 Coupling and Cohesion Metrics

#### 2.4.1.1 Number of Inputs/Outputs between components

Component	# Data/Events In	# Data/Events Out	Total
Authenticator			
User Handler			
Payment Handler			
Reservation Handler			
Search Handler			
Social Handler			

#### 2.4.1.2 Number of dependencies between components

Component	# Components to which this component sends or from which this component receives data/events
Authenticator	
User Handler	
Payment Handler	
Reservation Handler	
Search Handler	
Social Handler	

#### 2.4.1.3 Degree of Cohesion

Component	# Functions	# Functions that receive all inputs and send all outputs within component (not counting external)	% Functions that receive all inputs and send all outputs within component (not counting external)
Authenticator	3		
User Handler	3		
Payment Handler	9		
Reservation Handler	7		
Search Handler	6		
Social Handler	2		

#### 2.4.2 Size and Complexity Metrics

##### 2.4.2.1 Number of functions allocated to a component

Component	# Functions Allocated
Authenticator	3
User Handler	3
Payment Handler	9
Reservation Handler	7
Search Handler	6
Social Handler	2

##### 2.4.2.2 Number of data elements allocated to a component

Component	# Data Elements Allocated
Authenticator	2

User Handler	1
Payment Handler	2
Reservation Handler	4
Search Handler	1
Social Handler	2

2.4.2.3 Number of components in the blueprint

# Components in Blueprint
6

2.4.2.4 Component Complexity

Component	# Functions	# Data Elements	# Inputs and Outputs (across all functions)	Complexity
Authenticator				
User Handler				
Payment Handler				
Reservation Handler				
Search Handler				
Social Handler				

2.4.3 Support for Applied Heuristic