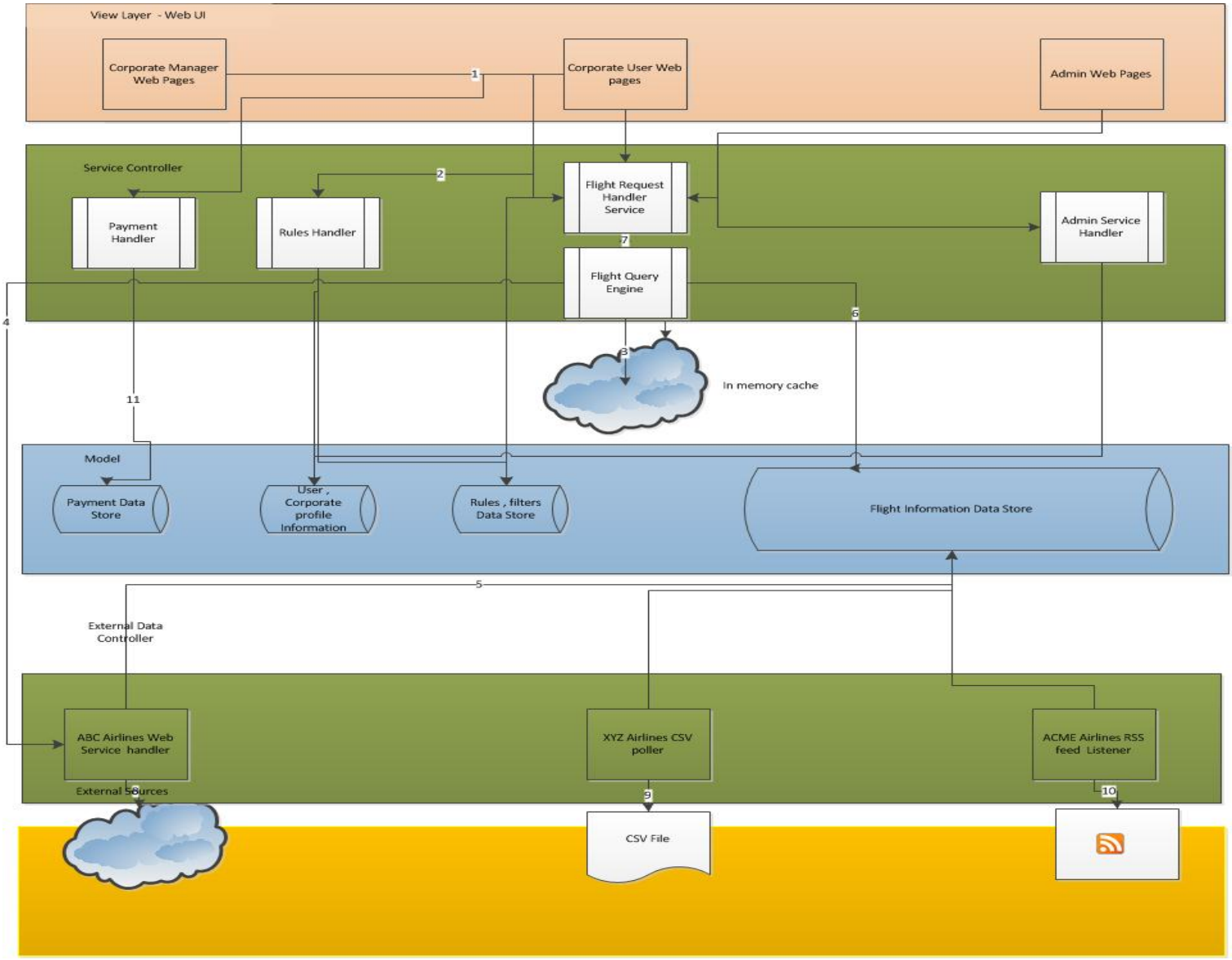
Tech Strategy Challenge

1.High Level Functional Architecture (please see attached in the folder for a clearer view)



2.Component Description

Sno	Component Name	Sub Component	Description of Functionality
1	Web UI		Ajaxian Multi Browser compatible web UI (JQUERY, HTML5)
		Corporate User	Query, Display

		Web Pages	results, Shopping cart, Payment)
		Corporate Manager web Pages	Ability to Add / Edit rules based to restrict what the corporate users can see. Another profile level based on the company can be added here as well
		Admin Web Pages.	Reporting and other Admin operations
2.	Service Controller		Native RESTful Web Service protocol that can hosted on any application server (RESTEasy)
		Flight Request Handler Service	Every incoming service request is routed via this web service and knows where to forward the appropriate incoming request. Performs multiple operations. 1. Introspects the user information on the request and gets appropriate rules / filters from Rules Handler 2. Send request to Query Engine and listens on Streaming Data from Query Engine (Can be implemented using NIO Sockets)

			3. Using AJAX, updates the Web UI as new data gets available.
		Flight Query Engine	Accepts incoming requests from the Flight Request Handler Service and is able to query the In memory Cache and return available results immediately. After a quick response from the cache, spawns a request to the ABC Airline Service Handler and waits on any newly updated data. Once received, returns to the Handler Service, Updates the Flight information DataStore and updates the local Cache.(Is Transactional)
		Rules Handler	Has Querying and Update operations. Received querying requests from Flight Request Service Handler based on user information. Updates / creates rules based on rules defined by Corporate managers
		Payment Handler	
		Admin Service Handler	
3.	In memory Cache		Able to define expiration policy and

			overflow mechanisms. Can be a vendor product or open source implementation. (Ehcache)
4.	Model	Name suggests the data the tables hold. 1. Flight Information data store 2. Payment data store 3. Rules data store 4. User Profile data store	Can be a relational (UDB, Sybase) or No-SQL Database (Cassandra) Further analysis needed to evaluate based on performance / cost factors
5.	External Data Controller		Feed handlers/ Data Access Objects into Flight Data Store /
		ACME Airlines RSS Feed Listener	A simple OpenSource feed listener which listens on an update every 30 mins. Updates the flight data store on incoming request
		XYZ Airlines CVS Poller	Will be an Autosys job that will be spawned everyday at 6 AM and updates the Flight information dataStore.
		ABC Airlines Web Service Handler	Is able to Service the requests from Query Engine and update the Flight information Data Store on demand.

3. Data Flow Description – ***Please refer the numbers on the arrows of the Functional diagram. They correspond to the serial numbers below.***

Sno	Source Component	Target Component	Description of Dataflow
1.	Corporate Manager Web Pages / Corporate User Web Pages	Flight Request Service Handler	
2.	Flight Request Handler Service	Rules Handler	Request Response.
3.	Flight Request Service Handler	Flight Query Engine	
	Flight Query Engine	In memory cache	
4.	Flight Query Engine	ABC Airline Web Service Handler	
5.	ABC Airline WebService Handler	Flight Info data Store	
6.	Flight Info data Store	Flight Query Engine	
7.	Flight Handler Service	Web Pages	
8.	ABC WebService Handler	ABC Company Service	Multiple Request Response
9.	XYZ CVS Poller	XYZ Company Service	One time / day Request Response
10.	ACME RSS Feed Listener	ACME Company	Continuous RSS Feed Listener
11.	Payment Handler	Credit Card Service	Request Response
12.	2 Sigma Admin Service	User Profile Information / Reporting	Request Reponse

4. Functional Architecture High Level Overview

WBS#	Description of Task	Skills Required	Estimated Man Days
1.0	Product Backlog requirements / Analysis	Requirements Analysis , UML	50

1.0.1	Planning & resource Allocation	Project management skills	50
2.0	Design	Technical Architect, Design Patterns,	120
2.0.1	Web UI	UI prototyping tools, Storyboarding, tools like Balsamique	40
2.0.2	Core model Design	IBM Data Architect	40
2.0.3	Core busisness Logic Design	Design Patterns	40
3.0	Development & Integration	Java	175
3.0.1	Infrastructure Setup	Scripting, Maven, Hudson	15
3.0.2	UI Development	HTML5, AJAX, XML, JAVASCRIPT	30
3.0.2	Core Controller / Web Services Development	RESTful service, RestEasy, Tomcat, Apache	40
3.0.3	External Source Handler / Data Access Objects	RSS handling mechanisms, SQL, Data Normalization	30
3.0.4	Datamodel Development for Data Store	SQL	20
3.0.5	Unit Testing	JUNIT	10
3.0.6	Integration		40
4.0	Testing		50
4.0.1	Functional Testing	Fitnesse	10
4.0.2	Performance Testing	Fitnesse	20
4.0.3	User	Manual	20

	Acceptance Testing		
5.0	Support & Documentation	Document Writers	

5. High Level Schedule – ***No time to answer this. However, the way I would approach this is using basic AGILE methodology using SCRUM. Will Release cycle will be a month 3 to 4 Sprints between each release.***

Sno	Key Date	Milestone	Dependencies

6. Key Risks and Risk Analysis

1.Technical

1.1 Is a packaged solution available? :

1.1.1 Ensure buy/build option is appropriately conducted.

1.2 Time required for learning curve during development.

1.2.1 Conduct training. Recruit experienced staff. Obtain vendor support.

2.Reliability/Availability

2.1 24 hour availability:

2.1.1 Increase time scheduled for physical design and construction stages. Involve technical experts in physical design stage.

3.Scalability/Performance

3.1 Database performance:

3.1.1 Increase data validation steps throughout. Ensure thorough Physical Design Tuning. Increase DBA involvement.

3.1.2 Attempt to establish overall volumes and requirements. Keep design flexible.

3.2 WebService / Website availability

3.2.1 Elongated prototyping phase to ensure handling of 3 times estimated volume is possible

4.Security of Customer and vendor Data

4.1Poor vendor support

4.1.1Impose contractual constraints/safeguards. Request documentation in advance. Ensure effective account manager. Identify a user group with other clients.

5.Legal / Contractual Agreements

5.1 Legal obligations regarding user data security

5.1.1 Involve Firmwide legal team early in the process

5.2 SLA agreements

5.2.1Document support procedures and agree the resources with management.

6.Vendors or Customer

6.1 Is a packaged solution available?

6.1.1 Ensure buy/build option is appropriately conducted.

6.2 Rework may be necessary, as system may not meet user's needs

6.2.1 Increase user involvement. Have multiple user touchpoints.

6.3 Unrealistic Expectations

6.3.1Schedule briefing and training sessions early in project. Increase user involvement/participation.

7.Others

7. Assumptions Made

1. RSS feeds are of a standard implementation
2. Database choice . Relational Vs NO-SQL