TLA Airline Agent

[Project Plan]

Version 1.0

By: Alec Smecher, Brent Whitman, Kun Wu, Linghong Liu, Lydia Qing Li, Mike Holley, and Tina Chen

Revision History

Revision Number	Date	Primary Author(s)	Comments
1.0	Oct 4th, 2001	Alec Smecher, Brent Whitman, Kun Wu, Linghong Liu, Lydia Qing Li, Mike Holley, Tina Chen	First version

Table of Contents

1 Introduction

1.1 Project Overview

The Airline Agent project is an intelligent agent providing assistance to users wishing to plan a flight. Given a user's current location, a location to which they wish to travel, and a number of constraints to further limit possible flights, the Airline Agent will provide a sorted list of possible tickets ordered based on the user's needs.

1.2 Project Description and Scope

The Airline Agent is part of a larger e-commerce project being undertaken by TLA's client. TLA has been contracted to fulfill the agent portion of the software, and TLA's client has agreed to perform the integration of the Agent component with the larger application. TLA has been given permission to perform data design and requirements research independent of the client's eventual e-commerce project.

The basis of the Airline Agent system is an intelligent search facility that takes a number of possible constraints into account. These can be such limits as:

- The user intends to leave from a specific airport within their current city;
- The user may arrive in any airport within a 100-kilometer (for example) radius of the desired city;
- The user intends to fly business-class;
- The user must arrive before November 17th;
- It is extremely important that the user not spend more than $900 on the ticket; and
- The user can permit one intermediate stopover.

These items are given as an example. Final possible constraints will be discussed in a later document.

Given these preferences, the Agent retrieves a list of possible flights, bearing in mind that the price and time of arrival are extremely important to the user.

As this project involves only the Agent, such aspects as user profiling, flight data input, authentication, and financial transactions are beyond its scope. These are the responsibility of TLA's client, as is the integration process with their product once it is completed. TLA shall provide the deliverables outlined in this document, along with database schema, a test data set, and all source code.

The agent is responsible for the ticket selection process once the users have successfully identified themselves. The agent's responsibilities are complete once a ticket has been selected, and the Client's e-commerce application is responsible for reservation, payment, delivery, confirmation, and other such processes.

The system shall include all necessary user interfaces to quickly and intuitively determine the user's requirements and preferences.

As the product involves aspects of a Geographical Information System (GIS), the Client may be required to purchase a detailed database containing the coordinates of cities containing airports, and the airports themselves.

2 Project Organization

2.1 Team Structure

The following is the list of executive roles, as required by CMM level 3.

- Senior Management: Alec Smecher
- Software Configuration Manager: Linghong Liu
- Software Engineering Project Group: Tina Chen, Lydia Li, Kun Wu
- Software Quality Assurance Engineer: Brent Whitman
- Representative of Customer: Mike Holley

2.2 Roles and Responsibilities

Project Manager: Alec Smecher

- Oversees project progress
- Approves and executes project plan
- Assigns tasks and reports status of project to team members
- Manages and motivates team members
- Represents the team to the outside world

Requirement Specification Analyst: Mike Holley

- Collects information from client interviews
- Develops concepts of the system for designers
- Creates requirement specification document

System Architect: Lydia Li

- Designs logical system based on requirements
- Translates logical design into detailed design
- Creates detailed design document

Usability Analyst: Tina Chen

- Designs User Interface
- Ensures stability and response time of the system meet the requirements
- Creates user manual

Software Configuration Engineer: Linghong Liu

- Integration of coded modules into functioning system

Developer: Linghong Liu, Kun Wu

- Implements product based on detailed design document

Quality Assurance Engineer: Brent Whitman

- Ensures acceptable software quality
- Designs testing strategies
- Creates and manages test plan
- Verify software requirements
- Executes test procedures

2.3 Team Communication

TLA communication channels include the following:

- Weekly meetings are held on Tuesdays.
- Group announcements and updates are sent through email: tla-475@sfu.ca
- Telephone discussions are held as necessary.
- Split up into subgroups as necessary, in order to work more co-operatively on specific problems.

3 Process Definition

3.1 Lifecycle Model

TLA intends to use the Incremental Development Model throughout the Airline Agent project. This methodology is more flexible than the traditional Waterfall SDLC due to repeated iterations involving design, coding, unit testing, integration, and quality assurance. The Waterfall SDLC is not a viable choice due to the short timeline available for the Airline Agent project to reach delivery quality.

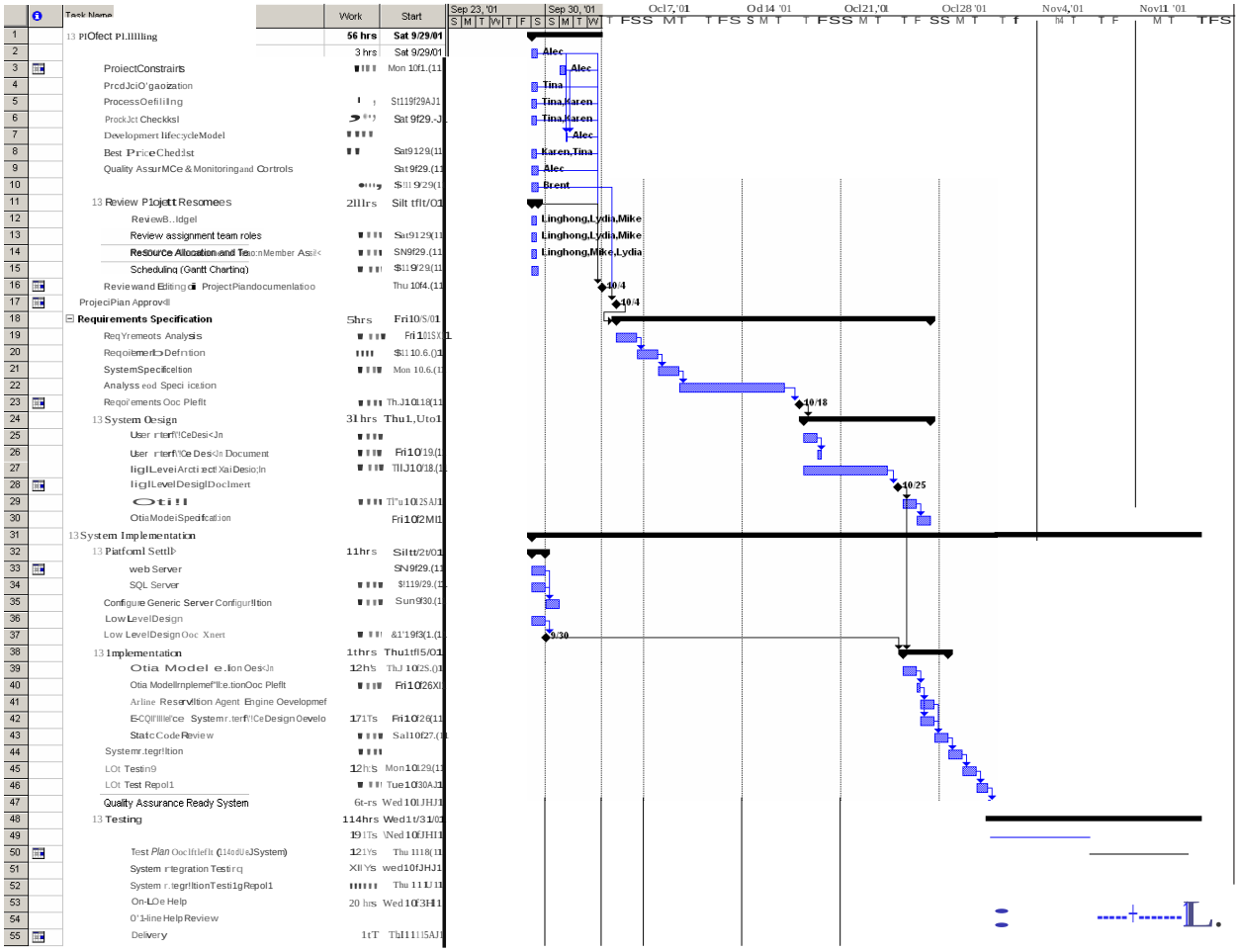
TLA has chosen to avoid such methodologies as Spiral because of concerns over the short timeline. Should design procedures, for example, need to be revisited within the first release date, it is likely that the project will overshoot its critical schedule.

TLA intends to deliver the first iteration of functionality on the System Delivery date indicated in the Estimations section of this document. After further client interaction, further iterations should occur as necessary.

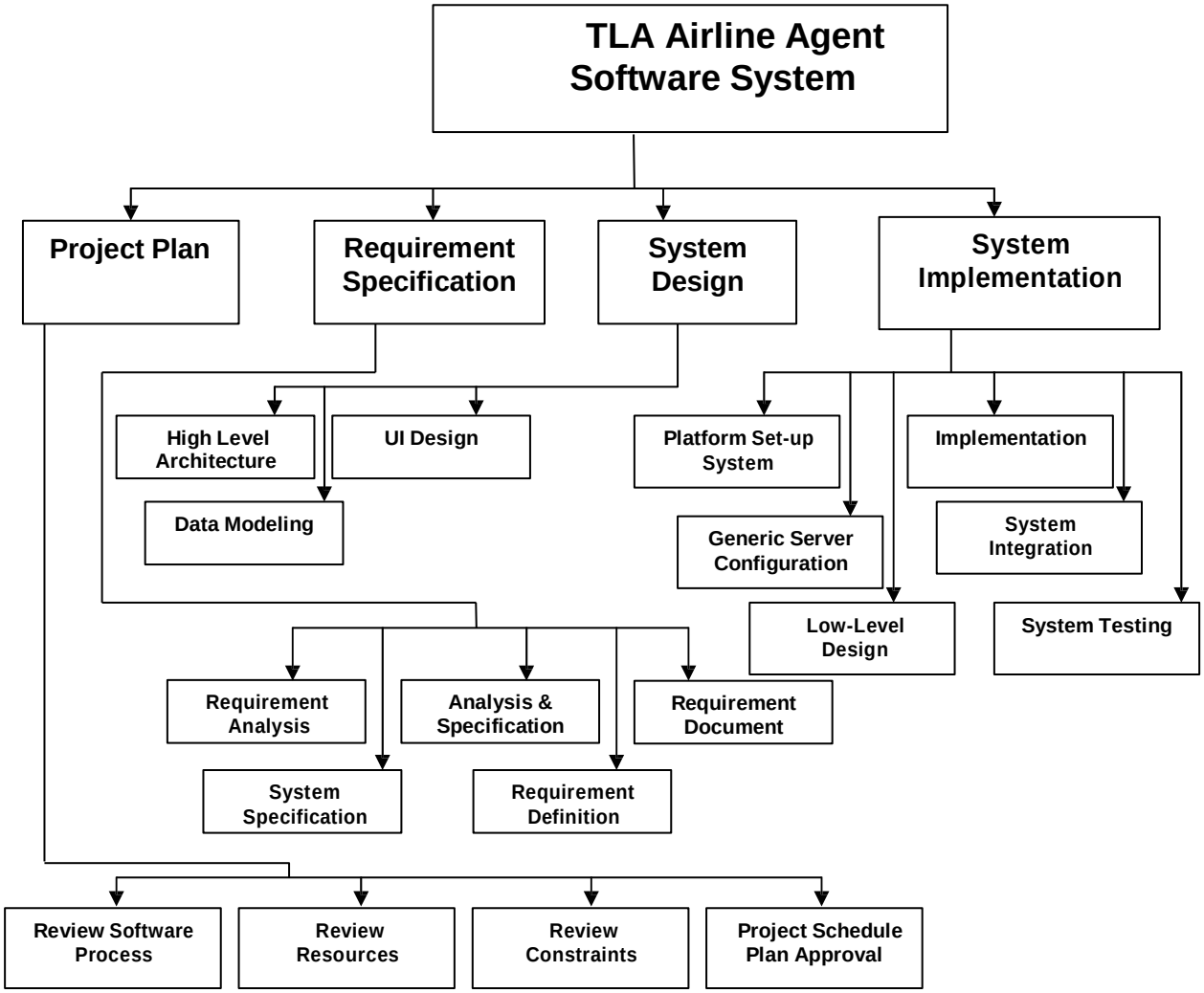
Due to the intelligent nature of this agent, further iterations interleaved with client interaction and live testing should hone the algorithms used such that they behave in an accurate and logical manner, providing a more effective search experience for users.

4 Schedule

4.1 Activity Dependencies and Schedule



4.2 Work Breakdown Structure



4.3 Work Packages

The entire project work is broken down by the important phases of the software development life cycle. They include the following:

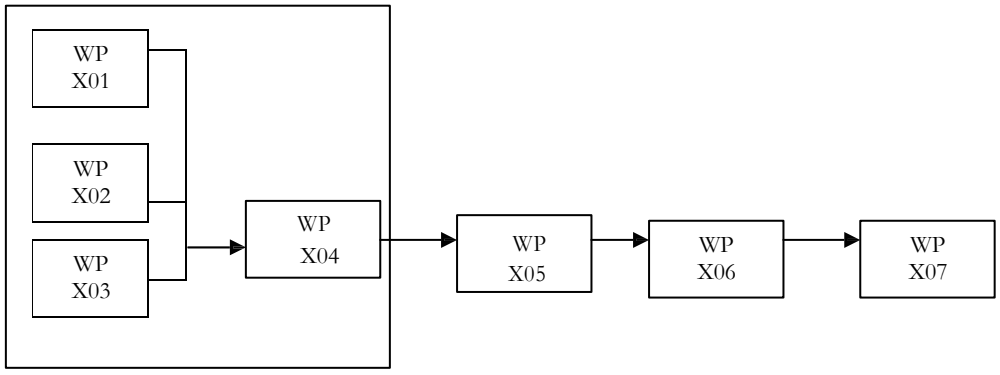
1. Project Plan
2. Requirement Specification
3. User Interface
4. Technical Architecture
5. Data Modeling
6. Coding & Unit Testing
7. Integration & Quality Assurance

4.4 Activity Dependencies

The following table describes the dependencies of the deliverable work packages:

Work Package #	Work Package Description	Duration	Dependencies
X01	Project Plan	7 days	--
X02	Requirement Specification	7 days	--
X03	User Interface	7 days	--
X04	Technical Architecture	12.1 days	X01,X02,X03
X05	Data Modeling	7 days	X04
X06	Coding & Unit Testing	16.2 days	X05
X07	Integration & System Testing	16.2 days	X06

The following Activity Network Diagram describes the above in more graphical detail:



Note that work package X05 is dependent on all work packages encapsulated by the larger boxes linked to its left. For instance, WP X05 may not start until WP X01- X04 has been finished.

4.5 Work Package Details

Work packages are listed below. A team member, indicated in bold, has been assigned as primarily responsible for each work package and will coordinate that package.

Project	TLA Airline Agent Software System
Work Package	X01— Project Plan (1 of 7)
Assigned To	**Brent Whitman**, Alec Smecher, Tina Chen, Lydia Li, Kun Wu, Mike Holley, Linghong Liu
Effort	7PD
Start Date	Saturday, 09/29/01
Purpose	To determine an introductory overview of the project, to be refined in later work packages.
Inputs	None
Activities	This work package includes providing a brief overview of the project, its objectives, and a set of proposed project deliverables throughout the development of the software cycle. The people responsible for this work package will also be transcribing ideas brought up in the group meeting discussion into a formal report.
Outputs	A written document of the Project Plan Introduction.

Project	TLA Airline Agent Software System
Work Package	X02— Requirement Specification (2 of 7)
Assigned To	**Mike Holley**, Brent Whitman, Linghong Liu, Alec Smecher, Tina Chen, Lydia Li, Kun Wu
Effort	7PD
Start Date	Thursday, 10/05/01
Purpose	To establish a common understanding between the customer and the software project team of the customers' requirements to be addressed by the project
Inputs	Customer's requirements
Activities	Identify "the customer", interview customer, write and inspect customer requirement and build requirements.
Outputs	A written document of the requirement specification.

Project	TLA Airline Agent Software System
Work Package	X03— User Interface (3 of 7)
Assigned To	**Tina Chen**, Lydia Li, Kun Wu, Mike Holley, Brent Whitman, Linghong Liu, Alec Smecher
Effort	7PD
Start Date	Thursday, 10/11/01
Purpose	To build the user interface between the system and the customer, to make it easy use, and friendly to the customer
Inputs	User information
Activities	To get the user information, user request, display the dialog between system and user, display the result of request
Outputs	User Interface

Project	TLA Airline Agent Software System
Work Package	X04— Technical Architecture (4 of 7)
Assigned To	**Lydia Li**, Kun Wu, Mike Holley, Brent Whitman, Linghong Liu, Alec Smecher, Tina Chen
Effort	12.1PD
Start Date	Thursday, 10/11/01
Purpose	To do the high level architecture design
Inputs	Project Plan Work Packages (X01 to X03 inclusive).
Activities	High level design entails defining the architecture of the software system and identifying the various components and how they are inter-related to and interactive with each other. Designers also need to decide on the software and hardware infrastructures, such as what operating system on which the software is built, the language used to implement the software, and so on. Design topics including maintainability, portability, and reusability will be addressed here as well.
Outputs	High Level Design and Architectural Specification.

Project	TLA Airline Agent Software System
Work Package	X05— Data Modeling (5 of7)
Assigned To	**Kun Wu**, Mike Holley, Brent Whitman, Linghong Liu, Alec Smecher, Tina Chen, Lydia Li
Effort	7PD
Start Date	Thursday, 10/25/01
Purpose	To build the project's database
Inputs	Project Plan Work Packages (X01 to X05 inclusive).
Activities	Analyze the data flow relationships, entity relationships
Outputs	A written document of the data modeling

Project	TLA Airline Agent Software System
Work Package	X06— Coding & Unit testing (6 of 7)
Assigned To	**Linghong Liu,** Alec Smecher, Tina Chen, Lydia Li, Kun Wu, Mike Holley, Brent Whitman
Effort	16.2PD
Start Date	Friday, 10/26/01
Purpose	To implement the system as per the requirements specification and other associated documents. This work package includes such additional activities as preliminary unit testing.
Inputs	Project Plan Work Package X06.
Activities	Programmers will implement the modules according to the design specifications noted in the Specification document.
Outputs	Source code and header files

Project	TLA Airline Agent Software System
Work Package	X07— Integration & System Testing (7 of 7)
Assigned To	**Alec Smecher,** Linghong Liu, Tina Chen, Lydia Li, Kun Wu, Mike Holley, Brent Whitman
Effort	16.2PD
Start Date	Sunday 10/28/01
Purpose	To identify and fix logical and syntactical errors produced during the implementation of the System, and setting up drivers and stubs to see how the module responds to various inputs. Black box testing as well as white box testing might be conducted to check for logical errors. All the testing procedures will be documented in the Test Plan report. If problems are found, they will be noted and fixed at the earliest possible time.
Inputs	Project Plan Work Package X07.
Activities	The Integration testing team may try to simulate how a user might interact with the system. Similar to Unit Testing, Integration Testing may require the development of stubs and drivers as well, but here this is more geared towards the higher (overall system) level. Testers may also examine issues such as system performance and integrity. Heuristics assessment plays an important role in this work package, as intelligence components will define eventual system success.
Outputs	A test report.

5 Project Estimates

5.1 Code Size Estimation using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

5.1.1 Unadjusted Function Points

TLA airline agent supports the following proposed functions:

Customer:

- Gather trip information from customer
- Collect and analyze customer preferences
- Provide sorted results based on customer's entered criteria
- Launch the reservations
- View/Modify/Cancel reservations in allowable period

Administrator:

- List customer preferences
- List customer reservations

The measure of unadjusted function points is based on five primary component elements of these functions: Inputs, Outputs, Inquiries, Logical Files, and Interfaces. Each element ranges from Low Complexity, Medium Complexity to High Complexity. The detailed evaluation of the complexity is as follows:

Rating Inputs:

- Gathering flight information: (departure and destination locations, departure and arrival times, email address, first name, last name, address, error message, confirmation message)
- Preferences (i.e. class, available meals) and Constraints (i.e. sorting the results by cheapest price, first available flight, airline company, airport location)

Files Type Referenced (FTR)	Data Elements		
	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

Rating Outputs:

- Displaying a list of the results matching the customer's search criteria (flight number, departure time, departure airport, airline company, class, price, arrival time, arrival location)
- Displaying customer reservations
- Displaying customer account information

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

Rating Inquiries:

- Selecting the flight according the customer's criteria
- Selecting customer reservations
- Selecting customer account information

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

Rating Logical Files:

- Flight Reservation
- Customer Account

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Rating Interfaces:

- 5 External Files Referenced (Cities, Airlines, Airports, Airplanes, Classes)

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

Summary of above analysis:

Element	Complexity	Detail
Inputs	Low	Gathering flight information
	Low	Preference and Constraints
Logical Files	High	Flight Reservations
	Medium	Customer Account
Outputs	High	Display Search Results
	Low	Display Reservations
	Low	Display Customer Accounts
Inquiries	High	Selecting the flight according the customer's criteria
	Low	Selecting customer reservations
	Low	Selecting customer account information
Interfaces	Medium	Cities, Airlines, Airports, Airplanes
	Low	Classes

Calculation of Unadjusted Function Points:

Characteristic	**Low**		**Medium**		**High**	
Inputs	2	× 3	0	× 4	0	× 6
Outputs	2	× 4	0	× 5	1	× 7
Inquiries	2	× 3	0	× 4	1	× 6
Logical Files	1	× 7	0	× 10	1	× 15
Interfaces	1	× 5	4	× 7	0	× 10
Unadjusted FP	32		28		28	
Total=L+M+H	88					

5.1.2 Adjusted Function Points

Influence Factors	**Score**	**Detail**
Data Communications	5	Application is more than a front-end, and supports more than one type of teleprocessing communications protocol.
Distributed Functions	4	Distributed processing and data transfer are online and in both directions.
Performance	3	Response time or throughput is critical during all business hours. No special design for CPU utilization was required. Processing deadline requirements with interfacing systems are constraining.
Heavily used	2	Some security or timing considerations are included.
Transaction rate	3	Daily peak transaction period is anticipated.
On-line data entry	5	More than 30% of transactions are interactive data entry
End-user efficiency	2	Four to five of the efficiency designs are included
On-line data update	3	Online update of major internal logical files is included.
Complex processing	1	Any one of the complex components
Reusability	4	The application was specifically packaged and/or documented to ease re-use, and the application is customized by the user at source code level.
Installation Ease	1	No special considerations were stated by the user *but* special setup is required for installation.
Operational Ease	1	Effective start-up, back-up, and recovery processes were provided, but no operator intervention is required (count as two items).
Multiple sites	0	User requirements do not require considering the needs of more than one user/installation site.
Facilitate change	3	Flexible query and report facility is provided that can handle complex requests, for example, *and/or* logic combinations on one or more internal logical files (count as three items).
Total score	37	
Influence Multiplier = Total score × 0.01 + 0.65 = 37 × 0.01 + 0.65 = 1.02		
Adjusted FP = Unadjusted FP × Influence Multiplier = 88 × 1.02 = 89.76		

Scoring (0 – 5)
0 = No influence
1 = Insignificant influence
2 = Moderate influence
3 = Average influence
4 = Significant influence
5 = Strong influence

5.1.3 Lines of Code

According to Capers Jones statistics, each Function Point requires 29 lines of code if the application is implemented using C++.

Therefore, we have: **Lines of Code** = 89.76 FP × 29 LOC/FP = **2603 LOC**

5.2 Efforts, Duration and Team Size Estimation

To estimate the effort and duration required for the project, we use both Top down and Bottom up estimation method, which provides validation to each other. This combination of schemes allows the team to arrive in a reasonable and practical schedule.

The estimates are expanded to account for project management and extra contingency time to obtain the total average effort estimates. From these averages, the duration of each work package in working days is estimated based on the following calculations:

5.2.1 Top down estimation

In this method, we use function points as the basis to calculate Effort, Duration, Team size, Compression rate and finally the schedule.

- Working days include 7 days in a week.
- Effort = Size / Production Rate = (2603 LOC) / (39 LOC/PD)[1] = 67 PD
- Duration = 3 × (Effort) $^{1/3}$ = 3 × (67) $^{1/3}$ = 12.2 PD
- Initial schedule = 67 PD / 7 days a week = 9.57 weeks
- Team size = 67 PD / 12.2 PD = 5.49 person
- But since we have 7 persons, the **Compression rate** = 5.49 person / 7 person = 0.785[2]
- **Desired schedule** = Initial schedule × Compression rate = 9.57 week × 0.785 = **7.51 weeks**
- **Total person-hours** = 67 PD × 8 hours = **536**

5.2.2 Bottom up estimation

In this method, we base our estimation on the following factors:

- Remaining semester length
- Balance of work load between other class
- Balance of time between studies, work and leisure

The calculation is as following:

- There are 7 group members; each spends 10 hours/week; then total hours per week is 70
- The remaining time of course schedule is 8 weeks, so the total person-hours is 560

[1] Lines of code per Person Day statistics based on Industrial Benchmarks, 1997: 31 LOC/PD for United States; 62 LOC/PD for Canada

[2] Statistics indicate that compression rates lower than 0.80-0.75 are not easily attainable; The calculated compression rate of 0.785 indicates that the timeline will be difficult but not unattainable.

5.2.3 Distribution of Effort

1990's Industry Data	Work Package	Distribution	Top-Down Estimates	Bottom-Up Estimates
Preliminary Design 18 %	Project Plan	9%	48.24	50.4
	Requirement Specification	9%	48.24	50.4
Detailed Design 25 %	User Interface	7%	37.52	39.2
	Technical Architecture	11%	58.96	61.6
	Data Modeling	7%	37.52	39.2
Code & Unit Testing 26 %	Code & Unit testing	21%	112.56	117.6
	Online Documentation	5%	26.8	28
Integration & Test 31 %	Integration & Quality Assurance	31%	166.16	173.6
	Extrapolated total effort		536	560
	2% for project management		10.72	11.2
	3% for contingency		16.08	16.8
	Total effort		562.8	588

These duration estimates are based on the assumption that each team member works an equal amount on any given work package

5.3 Cost Estimates

Hardware:

Developer workstations:

7 - Dell Precision Workstation 330	
Pentium IV 1.4GHz single processor	
256 MB RAM	Total
20.8GB IDE drive	$0.00

Software:

GNU, Apache, or Other Free License-based Software:

Apache Web Server	$0.00
Perl	$0.00

Software License Provided by Third Party:

Microsoft Office 2000	$0.00
Microsoft Project 2000	$0.00
Products ESTIMATE Professional	$0.00

Other Resources:

Staff:

7 Employees with 562.8 working hours with $18.00/hour	$10,130.40

Stationary:

Paper, photocopying and other miscellaneous cost	$50.00

Total:

$10, 180.40

The customer will supply the required hardware and software necessary to run the SQL server and the E-Commerce web server. TLA is not responsible in anyway for supplying said systems. TLA 's hardware and software responsibilities relate only to our own development needs to accomplish the project we have been asked to complete, and which has been described in the introduction section of this document. TLA will also demonstrate the completed product.

6 Product Checklist

The plan is that the items listed below will be delivered on the stated deadlines.

Project Deliverable	Estimated Deadline
Project Plan	Oct 4th, 2001
Requirements Specification	Oct 18th , 2001
Design Document	Oct 25th, 2001
Module/System Test Plan	Nov 8th , 2001
System Release (Demo)	Nov 15th, 2001

7 Best Practice Checklist

Practice	9
Document what we do; all documentation must be in a standardized format.	
Pay attention to requirements, check for ambiguity, completeness, accuracy, and consistency. The requirement documentation must contain a complete functional specification.	
Keep it simple. Complexity management is one of the major challenges. Strive to: • Minimize interfaces between modules, procedures and data. • Minimize interfaces between people, otherwise exponential communication cost • Avoid fancy product functions, design as long as the functionality meets the customer requirements	
Require Visibility. We must see what we build otherwise we can measure the progress and take management action. This includes: the manager must have good communication with his or her employees; require developers to make code available for review; review design for appropriateness.	
Plan for continuous change. We must: • All manuals designs, test, source code should have revision numbers and dates revision history comments, change marks to indicate the changes • New revisions should e approved before being made and checked for quality and compliance after being made • Use a configuration management system and make processes • Required maintenance	
Don't under estimate. We must be careful to obtain accurate estimates for: time, effort, overhead, meeting time, and especially effort on integration, testing, documentation and maintenance.	
Code reviews are a much more efficient method to find software defects. Plan and manage code reviews between team members	
Software testing will use both black box and white box testing. It will involve unit, functional, integrating and acceptance testing.	

8 Risk Management

Besides the general risk management, the following risks have been identified for the Airline Agent project:

More changes to requirements than anticipated
Impact Severity: High
Probability: 25%
Impacts: Depending on the stage at which changes occur, could range from needing to update the requirements documentation to a needing to do a complete redesign.
Risk Reduction: Be rigorous in eliciting requirements. Make customer aware of potential repercussions of requirement changes.

Specification Delays
Impact Severity: High
Probability: 15%
Impacts: Delay in finalizing the specification will push the schedule for all following stages of the project.
Risk Reduction: Monitor progress of specification carefully.

System size underestimated
Impact Severity: Moderate
Probability: 30%
Impacts: More work will need to be spent on design and coding; could negatively impact schedule.
Risk Reduction: Update estimates often as project progresses.

Staff leaving before project complete
Impact Severity: Extreme
Probability: 5%
Impacts: There would be more work for remaining employees, and any specialized skills or knowledge would be lost.
Risk Reduction: Offer benefits and incentives to staff.

Problems co-ordinating within group
Impact Severity: Moderate
Probability: 40%
Impacts: Members may be unaware of what is expected of them; managers may not be able to measure progress; portions of projects not completed.
Risk Reduction: Follow communication plans as documented in section 2.3

Customer cancels project
Impact Severity: Super-Extreme!
Probability: 1%
Impacts: All work will have been wasted.
Risk Reduction: Keep in close contact with customer. Ensure that they have some market research indicating a demand for this product.

9 Quality Assurance

The project will achieve the quality assurance by following the standard set by the company. The specific procedures and details shall be provided in the Quality Plan.

Specific test procedures and details shall be provided in the Module/System Test Plan.

In addition, the Airline Agent shall make use of two testing methodologies:

- **Unit Testing** involves testing system components individually.
- **In-Place Testing** involves testing of the whole system as a unit.

Furthermore, these methodologies will be used to test two important aspects of the Airline Agent:

- **System Function** will be tested to ensure that software flaws are eliminated, and
- **Algorithmic Function** will be tested to ensure that heuristic aspects of the project (such as user preference rankings) perform realistically to provide value to the users.

TLA's methodology makes broad use of realistic test cases. Detailed test data is an important part of the final project delivery. Although TLA's client is expected to furnish and enter data regarding flights, cities, airports, and the other aspects involved in the Airline Agent, TLA shall provide a comprehensive and detailed subset of this data for testing purposes. TLA will validate code and heuristic result ranking technology using realistic scenarios. In addition, extreme cases (such as departure from a city without an airport nearby) will be used to ensure that the system behaves correctly in degenerate cases.

10 Monitoring & Control

Many procedures are required in order to be able to successfully monitor the progress of a software project. Some of the most important are:

Quantitative measurement of resource consumption: Estimates of the Agent's resource requirements, primarily in terms of human resources, can provide a quantitative measurement of project progress when compared to progress in terms of project milestones. The percentage estimates of each milestone's resource requirements provided in this document allow for easy progress tracking.

Identification of major project risks: Early identification of major risks to the project allows for placement of preventative measures before problems can develop. Major risks have been identified in the Risk Management section of this document, along with the measures being taken to avoid them.

Regular reviews of project progress: Throughout the duration of the Airline Agent project, TLA shall meet weekly to review the progress of all project tasks, including management, planning, analysis, development, and testing.

Timeline Planning and task decomposition: This document outlines an estimated timeline for the project. A reasonably accurate timeline can be assembled by hierarchically decomposing tasks into measurable subcomponents and estimating requirements for each. At the same time, this decomposition can assist in task assignment and balancing. Throughout the implementation phase, these subcomponents can allow for fine-grained measurement of progress. Project subcomponents and timeline estimates are included in the Estimates and Work Breakdown Structure sections of this document.