Travel Companion

# SOFTWARE ENGINEERING PROJECT

## BY:

NANDITA SHUKLA 16BCE0553

MEENAKSHI DAS 16BCE0560

ABINASH SATAPATHY 16BCE0081

## GUIDED BY:

DR SHASHANK MOULI SATAPATHY

# TABLE OF CONTENTS:

1. SRS DOCUMENT

2. PROCESS MODEL IDENTIFICATION

3. WBS

4. ACTIVITY NETORK

5. CRITICAL PATH IDENTIFICATION

6. GANTT CHART

7. ER DIAGRAM

8. DFD-DATA DICTIONARY

# SRS DOCUMENT:

1. **FUNCTIONAL REQUIREMENTS**
   1. Create Trip

When an authenticated user logs in from their account or when a person signs up they get an option to create a new trip .The user has to select:

* the vehicle type
* the number of passengers for the journey
* the date of the journey
* the source
* the destination.

This function is initiated after login/signup after the user selects the create trip option from the user window.

The user that creates a new trip becomes the trip admin for the trip.

* 1. Request for join

When the user logs in , he/she is directed to the user window wherein if he chooses the option for ‘request for join’ , he gets a detailed view of the existing cabs for the trips that other users have already created.

1.2.1 - The user can then send a request to join an existing trip to the trip admin.

1.2.2 - The user can sort the search for the existing cabs by the source, destination or the date of travel.

1.2.3 - Request Response-The user window initiates another function i.e. the request response which has two primary sub-requirements:

1.2.3a-For a trip that the user acts as an admin, the user can accept or decline the requests for other users based on their details.

1.2.3b-For a trip that the user has requested , the status (accepted/pending/declined) can be viewed by the user.

1.3) Contact sharing

The users can view the contact details of the admin and vice-versa if the contact sharing option is enabled by the users for a certain trip or for everyone.

1.3.1Confirmation Message-The user gets notified if his cab is cab request gets confirmed .The confirmation is through email and sms to the email address and the contact number provided respectively.

1.4) Status of the cab

The users can see the location of the cab right now .This functional requirement tracks the gps of the cab.

1.5) Travel agency:

The users can choose from which travel agencies they want the car by analyzing the availability of cars and different rates.

**2. NON-FUNCTIONAL REQUIREMENTS**

2.1 System should accept payments via payment methods that are reliable like paypal, wallets ,cards, vouchers etc.

2.2 User during sign up , should be helped appropriately to fill in the mandatory fields in case of invalid input.

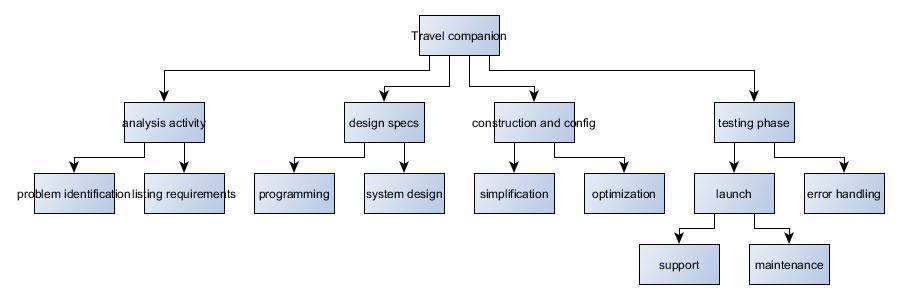
2.3 System should visually confirm as well as send booking confirmation to user’s contacts .

# PROCESS MODEL IDENTIFICATION:

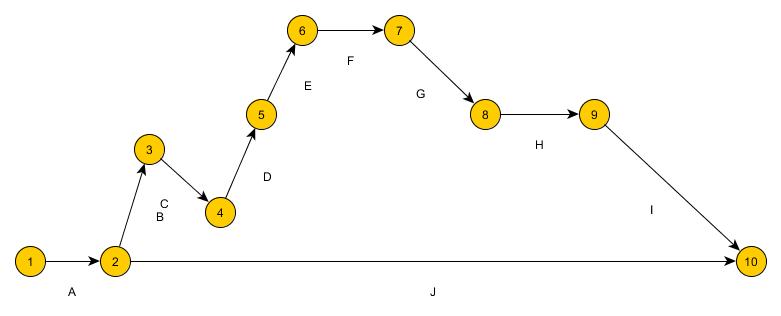
The model we are planning to use is **Sashimi** model (a variation of waterfall model). We are planning to use this traditional model because:

* The requirements are very well known
* The specifications are very well translated from the requirements
* It is a predictive model because we know the solution very well
* It is a waterfall model that shortens the development type by overlapping certain phases (since we have less time for development)

# WBS (WORK BREAKDOWN STRUCTURE):

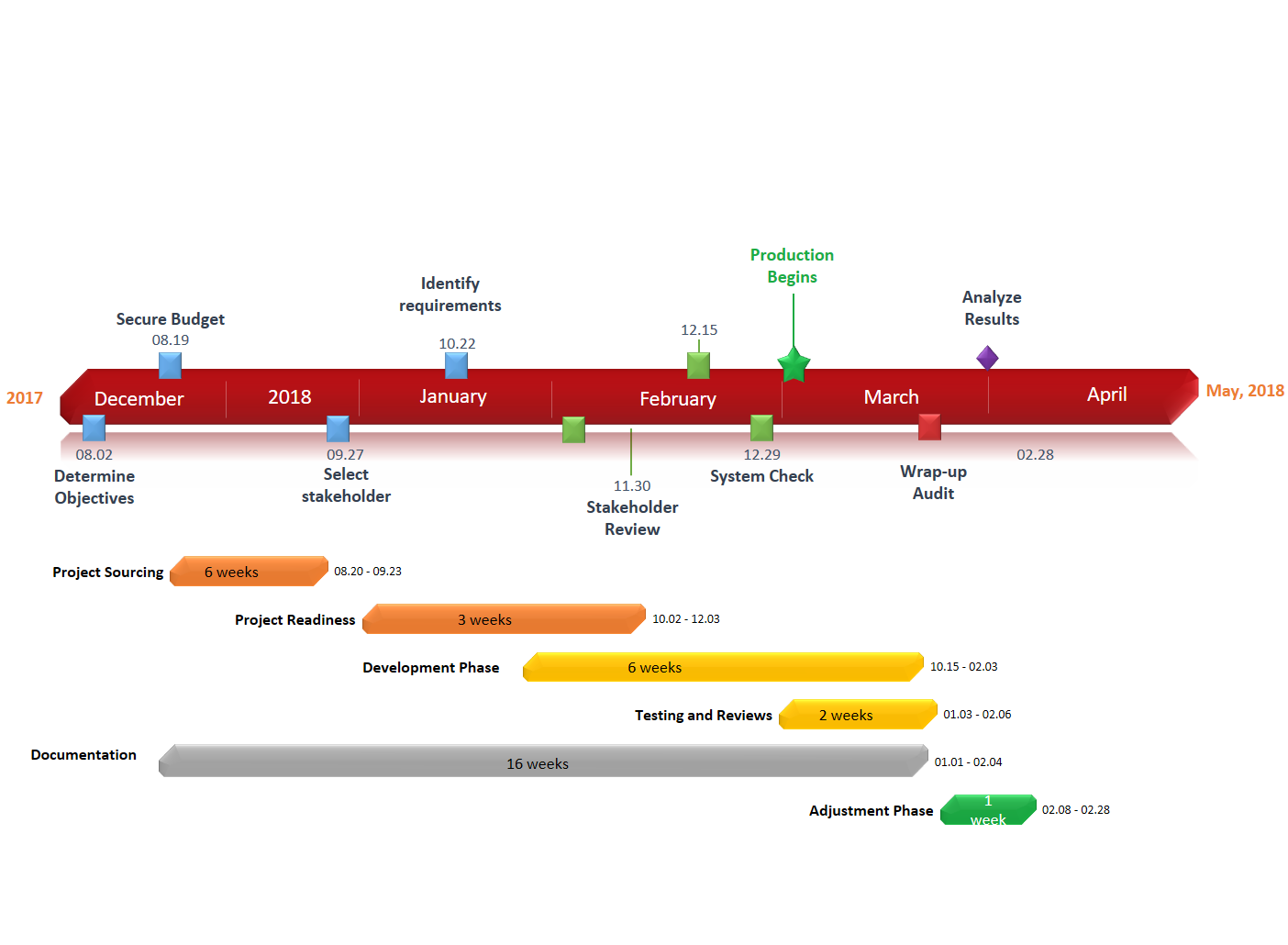


# ACTIVITY NETWORK:

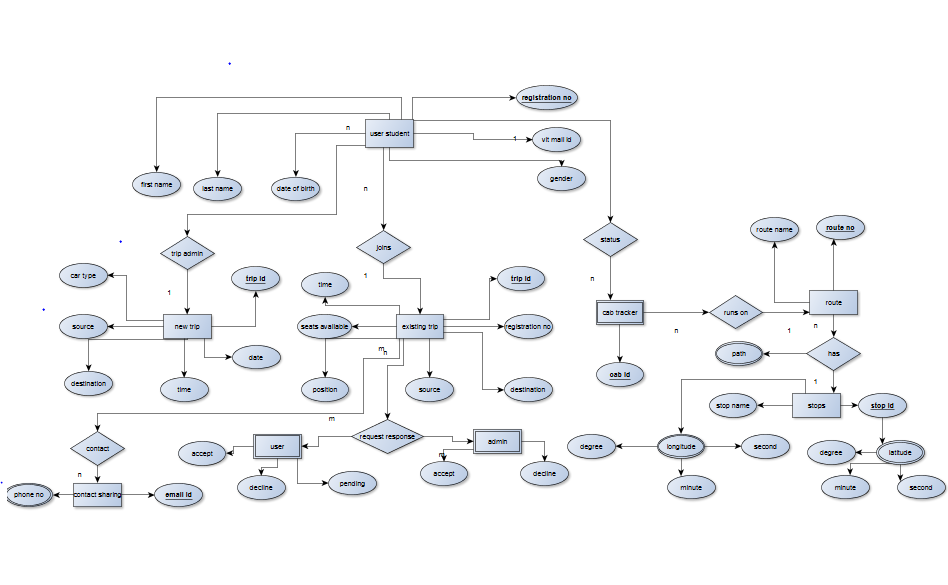


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ACTIVITY |  |  | DURATION |
| A | SET UP ACQUISITION TEAM | | | 1 week |
| B | WRITE DOWN SOFTWARE REQUIREMENTS | | | 3 weeks |
| C | DETERMINE OBJECTIVES | |  | 1 week |
| D | SECURE BUDGET | |  | 1 week |
| E | SELECT STACKHOLDER | |  | 1 week |
| F | STACKHOLDER REVIEW | |  | 1 week |
| G | SYSTEM CHECK |  |  | 2 weeks |
| H | PRODUCTION |  |  | 4 weeks |
| I | WRAP UP AUDIT |  |  | 1 week |
| J | ANALYSE RESULTS | |  | 1 week |

# GANTT CHART:



# E-R DIAGRAM:



# DATA DICTIONARY:

User name: Alphanumeric \* input to be taken \*

Password: Alphanumeric + Special characters \* input to be taken \*

Error: String \* Error message to be displayed (preferably a pop-up) \*

Source: String \*Input – name of a city \*

Destination: String \* Input – name of the city Destination ≠ Source \*

Vehicle type: String

Nop: Integer \* Number of passengers to be entered \*

Doj: Date format \* Date of journey \*

Contact: Integer \* Phone number of user \*

Email: Alphanumeric

# DATA FLOW DIAGRAM:

