|  |  |
| --- | --- |
|  | **App**  **Business Requirements Document (BRD) – Release 1** |

|  |  |
| --- | --- |
| **Authoring Group:** |  |
| **Contact Information:** |  |
| **Division/Business Unit:** |  |
| **Region:** |  |
| **Sub-Region:** |  |
| **Release:** |  |

**Table of Contents**

[1. Change History 3](#_Toc378092170)

[2. Introduction & Business Goals 4](#_Toc378092171)

[2.1. Intended Audience 4](#_Toc378092172)

[2.2. References 4](#_Toc378092173)

[2.3. Key Stakeholders 4](#_Toc378092174)

[3. Scope Summary 5](#_Toc378092175)

[3.1. In-scope 5](#_Toc378092176)

[3.2. Out of Scope 5](#_Toc378092177)

[4. Constraints and Assumptions 6](#_Toc378092178)

[5. Process Flow - Overall 7](#_Toc378092179)

[5.1. App Interface Diagram 7](#_Toc378092180)

[5.2. App Functional Flow 7](#_Toc378092181)

[5.3. App UI 8](#_Toc378092182)

[6. Business Requirements 12](#_Toc378092183)

[6.1. Problem Description 12](#_Toc378092184)

[6.2. Solution Description 12](#_Toc378092185)

[6.3. High Level Functionality of App 12](#_Toc378092186)

[6.4. User Management 14](#_Toc378092187)

[6.5. Database / Server Functionality 14](#_Toc378092188)

[6.6. Business Rules & Opportunities 16](#_Toc378092189)

[6.7. Application - Process Flow 17](#_Toc378092190)

[7. Other Functionality 18](#_Toc378092191)

[7.1. Reporting Requirements 18](#_Toc378092192)

[7.2. Admin Requirements 18](#_Toc378092193)

[7.3. Audit Trail 18](#_Toc378092194)

[8. Non Functional Requirement 19](#_Toc378092195)

[8.1. Application Performance 19](#_Toc378092196)

[8.2. Server Support 19](#_Toc378092197)

[8.3. Device /Operating System Support 19](#_Toc378092198)

[8.4. User Interface – General Requirements 19](#_Toc378092199)

[9. Appendix 20](#_Toc378092200)

[9.1. Product Definition 20](#_Toc378092201)

[9.2. Document Management – Templates 20](#_Toc378092202)

[9.3. Data Matrix 20](#_Toc378092203)

# Change History

**The following important changes have been made to this document:**

| **Version No.** | **Date** | **Description** | **Author** | **Comments** |
| --- | --- | --- | --- | --- |
| 0.1 | 20/1/2014 | Draft | Amit Shroti |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction & Business Goals

All attempts have been made in using mostly business terminology and business language while describing the requirements in this document. Very minimal and commonly understood Technical terminology is used.

## Intended Audience

The main intended audience for this document are the business owners, stakeholders and development team of the proposed system.

## References

|  |  |
| --- | --- |
| References | Location |
|  |  |
|  |  |
|  |  |
|  |  |

## Key Stakeholders

|  |  |
| --- | --- |
| Name | Title/Role |
| Arvind Gupta | Technical Architect, Developer, Tester |
| Sachin Bajpai | Solution, Sales & Marketing |
| Vikrant Jain | Developer, Tester |
| Amit Shroti | Project Manager |

# Scope Summary

## In-scope

* Create an App as per the given requirement, implementation and support plan
* Simple App with minimum installable space requirement
* Simple graphics (if any) so that App should be able to execute efficiently on handset with basic graphic cards
* Create a plan for sale and market
* Identifying Server location and space requirement
* Deal with legal compliance issues for App launch (if any)
* Deal with license of App (if any)

## Out of Scope

This section describes Out of Scope items for Release 1

# Constraints and Assumptions

* App will be available in English Language only
* App will be functional only for few Indian Cities initially.
* App will be Android based
* App should be simple enough for user to key-in his travel details within 60 seconds (maximum).
* App will require one time user registration and ‘Signing a legal terms and condition’.
* Search functionality to be provided for “Current location and End location” fields.
* User should able to download this App and install without any difficulties for irrespective of the handset configuration. However we can plan for further release and support of the App in future.
* User should able to uninstall (remove) this App from his mobile handset (if willing)

# Process Flow - Overall

## App Interface Diagram

App 1

Server

App 2

App 3

App n

## App Functional Flow

App

**Server / Local Database**

* **User information**
* **Travel Details**
* **Contains local data**
* **Data clean up on regular interval**
* **Access to Travel details by other user**

**User 1**

**User 2**

App

App

**User n**

## App UI

**On time ‘User Registration and Acceptance of Legal wordings”**

Screen TBD

User Registration involves:

1. Confirmation to use his mobile number to identify and connect
2. Select his current City (no need to enter, will get info from his/her device current location when app starts)
3. End location will be predefined for current city and option to edit it if it does not exist in predefined list
4. At the time of registration user has to enter few personal info e.g. name, age, sex, profession etc.
5. Start location will be by default his/her current location and option to edit it with predefined list.
6. Check status screen is not needed, there will be one screen with his/her current location and travel details. When user submit this data then server will perform search operation and will share search result with both users together in other screen and let them call to each other and manage their travel. When user calls to one person and due to some reason they can’t travel together then they must have other users info available at the hand.
7. I am confused with cancel and confirm option, here what do we want to achieve?
8. When user submit their travel info then it is not mandatory that he/she will get info immediately as it is possible that he/she is the first person to travel in that route for mentioned time and no info is present in server, so in that case he/she will receive info as another travel details info come to server for same route.
9. As user submit their data then server performs seach operation immediately and if data found then immediately shared that info with bot users.
10. As mentioned above if user is first person to travel in that route for mentioned time then he/she will get meaningful response having info something like “u will receive co-passengers info immediately when it will be available with us”.
11. When two user manage to travel with each other then he/she has to tell us (we need to find out better way to achieve this).
12. We may give options to share his feedback and travel experience with particular user at the end of travel which will be very useful for us and data analytics point of view as well.

**Home Screen for every new entry**

Name:

Current Location (Area): (Predefined)

Starting Location: (user defined)

End Location: (predefined)

Mode of travel: (predefined)

Starting time: hh:mm AM/PM

Total Person: (numeric, user defined)

Send

**Example for Home Screen for every new entry**

Name: Amit

Current Location: Magarpatta City

Starting Location: DC

End Location: Railway Station

Mode of travel: Auto

Starting time: 07:00 PM

Total Person: 1

Send

**Check Status: (Assuming there is single entry for this travel route)**

**Cancel will allow user to delete his Travel Details**

Name: Amit

Current Location: Magarpatta City

Starting Location: DC

End Location: Railway Station

Mode of travel: Auto

Starting time: 07:00 PM

Total Person: 1

Cancel

**Check Status: (Assuming there are multiple entries for his travel route)**

**Cancel will allow user to delete his Travel Details**

**Confirm will allow user to connect to other user**

Name: Amit

Current Location: Magarpatta City

Starting Location: DC

End Location: Railway Station

Mode of travel: Auto

Starting time: 07:00 PM

Total Person: 1

Name: Sanjay

Current Location: Magarpatta City

Starting Location: North Gate

End Location: Railway Station

Mode of travel: Auto

Starting time: 07:10 PM

Total Person: 1

Name: Nidhi

Current Location: Magarpatta City

Starting Location: Jasminium

End Location: Railway Station

Mode of travel: Auto

Starting time: 06:45 PM

Total Person: 1

SCROLL

Cancel

Confirm

Confirm

**On ‘Cancel’**

User’s travel details will be deleted and will not be allowed further access the list.

**On ‘Confirm’**

As soon as one user Confirm (connect to) the Travel Details of other user, Name & Contact information (Mobile no.) will be displayed to each other’s screen. So that they can connect and share their travel.

Screen TBD

# Business Requirements

.

## Problem Description

People often travel for their day to day needs from one place to another. For their short travel (within city) they often prefer private communication medium (like Auto, Cab, Taxi or private vehicle). Most of the time this type of travels are quick and couldn’t be plan in advance.

Growing number of vehicles on the road leads to traffic, environmental harms, fuel and money wastage.

Some of them would like to share their travel with similar travelers traveling the same destination, however with life being busy and time constraint it wouldn’t be possible for one to connect to other.

## Solution Description

User-friendly and simple to use Android based App will be designed to communicate between users, share their travel details with other interested travelers (user), travelling to same destination. This App could be used on the go (i.e. while travel) or before the start of travel (maximum 1 hour\* in advance).

Within few inputs user will be able to post (send) his travel details and will be able to access the existing list of travelers on the same route. On confirm, user will be able to view the contact details of other user and then both the entries (travel details) will be marked unavailable.

[Main Idea is to create an easy to use App with fast response time.

Assuming user key-in time for his travel details as 60 seconds, then in next 30 seconds he should able to get others user’s ‘travel details’ list, from which he can confirm and connect to preferred user in next 30 seconds. Making the total turnaround time less than or equal to 2 minutes]

App functionality for this release will be limited to connect the interested (willing to share) user traveling the same location, by sharing their mobile numbers to each other.

## High Level Functionality of App

High level functionality planned for App:

**One Time Registration**

* One time registration should ask user to
  + Select current ‘City’ [We are using City as a onetime input from user, so that he need not to enter each time while key-in Travel Details. If he wants to operate in other city then he needs to perform One Time Registration again, selecting the new city]
  + Accept the legal terms and condition [Will contain read only data. Format yet to decided]
* Based on user current city selection the Predefined fields will be populated for ‘Current Location’ and ‘End Location’.
* Search functionality to be provided for ‘Current Location’ and ‘End Location’.

Example, If user city = Pune then

Current Location and End Location = List of predefined ‘Areas’ in Pune City

**Travel Details**

* User access to App
* User key-in his travel details
* User able to access the other users travel details
* User able to connect the preferred (with same Current and End Location) travelers (user)
* User able to delete his travel details before someone connects (confirm) to him.
* User should not be able to modify his travel details. Alternately, For any modification he has to first delete his existing entry and create a new one then. [As user cannot be traveling to two different places at the same time]

**Other Fields**

* ‘Starting Location’ is user defined field. It represent the pickup point or starting point for user
* ‘Stating Time’ is user defined field. User could able to plan ranging from current time to 1 hour in advance. [As the App functionality is to share unplanned or quick travels within city]

Example. If current time is 6:00 PM. Then User can plan starting time as

* + Starting time: 7:00 PM (Maximum)
  + Starting time: 6:00 PM (Minimum)
  + Starting time: 7:15 PM (Not allowed)
  + Starting time 5:45 PM (Not allowed)
  + Starting time: 6:30 PM (Allowed)
* “Mode of Travel” should be a predefined list containing Auto, Cab, Taxi, Private Vehicle only.
* “Total Person” filed should have following constraint on “Mode of Travel”

|  |  |
| --- | --- |
| **Mode of Travel** | **Total Persons (maximum)** |
| Auto | 3 |
| Cab | 4 (not including driver’s seat) |
| Taxi | 4 (not including driver’s seat) |
| Private Vehicle | None (as capacity of private vehicle varies) |

* Server (Local Database) will also match and constraint the ‘total persons’ fields so that the total maximum capacity should be EQUAL or LESS between the two user and display those travel details only.

Example 1. If Abhay is traveling in Auto with total person = 2. At the same time Sumit wants to travel in Auto with total person = 2 then it will be consider as NO MATCH and should not be displayed to each other. As the maximum person limit for Auto is 3.

Example 2. If Abhay is traveling in Private Vehicle with total person = 2. At the same time Sumit wants to travel in Private Vehicle with total person = 2 then it will be consider as MATCH and should be displayed to each other. As the maximum person limit for Private Vehicle is NONE.

**Others**

* User should allow to confirm and connect to one user only and then both the entries (travel details of both the user) will be marked unavailable.
* In ‘Check Status’ screen user entry should always be the first entry on the screen.

Example. If there are 3 entries (travel details) on screen, then no. 1 should always be his entry (travel details) followed with other two entries (travel details).

* User should not able to create multiple new entries (travel details).
* Only upon deleting the existing one or unavailable or auto deletion by server, User will be able to create another new entry.

Example. If user1 is travelling from A to B. After reaching B he wants to travel to C or to A then in this case he has to delete he previous entry (travel details) before creating new one.

* System will be auto deleting the un-fulfilled entries (travel details) after 15 min of starting time.

Example: Starting Time: 7:00 PM will be auto-deleted by 7:15 PM if remain unfulfilled

Starting Time 6:45 PM will be auto-deleted by 7:00 PM if remain unfulfilled

## User Management

* User will be allocated space in Local database using a unique ID (may be mobile no.)
* User will be able to create (new entry) to local database with his travel details
* User will be able to delete his entry from local database
* User will be able to ‘save’ or ‘access his historical travel details’ (Out of Scope for this release)
* User1 will be able to share his GPS (Global Position System) with user2, so that it will be easy for the user2 to track user1. (Out of Scope for this release)

## Database / Server Functionality

**Note: This feature/implementation subject to change as per Technical Architect view.**

**Local database:** Every city will be divided into Area and each Area will have a local database (logical).

Example, Pune City will have following areas Magarpatta City, Hadapsar Gaon, Hadaspsar Ind Area, Kothrud Paud Road, Kothrud Karve Rd etc)

Server will be logically distributed based on City – Area. As soon as User 1 enter his travel details with his current location, a local database should be created. Every further entry for same Current Location will be residing in same Local database. Local database will be deleted with the removal of last entry (travel details).

[Idea is not engage complete server to match the travel details with all existing entries.

Also assuming, if App is being highly used by user at Railway Station, Airport and Magarpatta City for Pune, then why to allocate and engage space for each area of City.

Also it will be fast retrieval of entries from local database.]

Creation of local database will help fast retrieval and access of entries (travel details) to user.

New user should able to access and view all the existing entries (having same Current and End location) in local database and should able to connect (confirm) to preferred user. Once acknowledge both the user entries (travel details) to be marked as BLOCKED and will be made unavailable (TBD – Should we remove the entries on the screen or should we marked as Unavailable, to be discuss and clarify).

On initial phase the App will be limited to connect and share travel details among interested user if they have same Current and End Location. Current and End location are predefined Areas of City. However the user should able to provide his pickup point of his choice against Starting Location field.

User could able to key-in his travel details before 1 hour\* (maximum) of his travel schedule.

Entries will be BLOCKED and unavailable to others if both users connect with each other.

Entries will be marked DELETE if the current time > (starting time + 15 mins\* ).

**Note:** Audit log will capture every entry and exist to server (local database).

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Database Field** | **Comments** |
| 1 | Mobile no. | 10 digit mobile number |
| 2 | Name | 20 Chars |
| 3 | Current Location | Predefined Entry based on City |
| 4 | Starting Location | Pickup point (user defined) |
| 5 | End Location | Predefined Entry based on City |
| 6 | Mode of travel | Auto, Cab, Taxi and Private Vehicle |
| 7 | Starting time | hh:mm AM/PM |
| 8 | Total Person | Numeric (user define) |
| 9 | Status | Open, Blocked, Deleted |
| 10 | Audit date and time | Timestamp |

**Example of Local Database**

**Note:** Local Database will contain all the entries (travel details) for same Current Location (Area).

User should only be able to view entries (travel details) with matching Current Location and End Location of other user.

Example. In below entries (travel details), Amit should able to see only 3 entries including his.

And Zaid should able to see only 2 entries including his as Pritam and Jayesh entries (travel details) are Deleted and Blocked respectively.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mobile no.** | **Name** | **Current Location (Area)** | **Starting Location** | **End Location** | **Mode of travel** | **Starting time** | **Total Person** | **Status** | **Audit date and time** |
| 9850898508 | Amit | Magarpatta City | DC | Railway Station | Auto | 7:00 PM | 1 | Open |  |
| 9850198501 | Sunjay | Magarpatta City | North Gate | Railway Station | Auto | 7:10 PM | 1 | Open |  |
| 8800880088 | Nidhi | Magarpatta City | Jasminum | Railway Station | Auto | 6:45 PM | 1 | Open |  |
| 8800880089 | Arun | Magarpatta City | South Gate | Kothrud Paud Rd | Auto | 7:30 PM | 1 | Open |  |
| 8800880090 | Ashit | Magarpatta City | South Gate | Nigdi | Cab | 7:40 PM | 1 | Open |  |
| 8800880091 | Priyanka | Magarpatta City | DC | MG Road | Taxi | 7:10 PM | 1 | Open |  |
| 8800880092 | Zaid | Magarpatta City | Heliconia | MG Road | Auto | 7:10 PM | 1 | Open |  |
| 8800880093 | Pritam | Magarpatta City | No Entry | MG Road | Auto | 7:00 PM | 1 | Deleted |  |
| 8800880094 | Jayesh | Magarpatta City | aa | MG Road | Auto | 7:15 PM | 1 | Blocked |  |

## Business Rules & Opportunities

This section can be explored later to define Business Rules.

This section can be explored later to find the opportunities that could help to make this App better and use user information (including travel details) for business benefits.

## Application - Process Flow

Other existing ‘Travel Details’ in Server with same Current & End Location

Start

No

Yes

Yes

No

Yes

No

No

User 1 to view other User’s Travel Details including his Travel Details

End

Mobile number of both the user will be displayed to each other.

And their Travel Details will be unavailable further.

User 1 confirm (connect to) other User’s Travel Details

End

User 1 delete his Travel Details

Key-in Travel Details

User 1 able to view only his Travel Details

Server / Local Database

User 1

# Other Functionality

This section describes business requirements not captured in the preceding section.

**Note: Below requirement to be discussed and finalized with Technical Architect.**

## Reporting Requirements

**TBD**

## Admin Requirements

**TBD**

## Audit Trail

Audit log will capture entry, modification, deletion, status change for each entry or exist of travel details on server (local database).

Format TBD.

# Non Functional Requirement

**Note: Below requirement to be discussed and finalized.**

## Application Performance

| **#** | **Parameter** | **Details** | **Comment** |
| --- | --- | --- | --- |
| 1 | Response Time |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

## Server Support

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Server Name** | **Version** | **Comment** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |

## Device /Operating System Support

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Device Name** | **Operating System** | **Comment** |
| 1 | Apple iPhone 4s, 5, 5s |  |  |
| 2 | Samsung S3 |  |  |
| 3 | Samsung S4 |  |  |
| 4 | Apple iPad 2 |  |  |
| 5 | Samsung Tab |  |  |
| 6 |  |  |  |

## User Interface – General Requirements

| **#** | **Parameter** | **Details** |  |
| --- | --- | --- | --- |
| 1 | Tooltips |  |  |
| 2 | Error Messages |  |  |
|  |  |  |  |

# Appendix

## Product Definition

## Document Management – Templates

## Data Matrix