Top Level Design

|  |  |
| --- | --- |
| Proposal | Presentational prototype |
| Author/s | Niv Ben-David & Michael Dimenshtein |
| Version | 0.3 |
| Created | 17/11/2011 |
| Last Update | 19/12/2011 |
| Current Status | First draft |

Contents

[Revision History 3](#_Toc309331139)

[1 Introduction 4](#_Toc309331140)

[1.1 Overview 4](#_Toc309331141)

[1.2 Design Goals and Non-goals 4](#_Toc309331142)

[1.2.1 Goals 4](#_Toc309331143)

[1.2.2 Non-goals 4](#_Toc309331144)

[1.3 Dependencies, Assumptions and Design Constraints 4](#_Toc309331145)

[1.3.1 Dependencies 4](#_Toc309331146)

[1.3.2 Assumptions 4](#_Toc309331147)

[1.3.3 Design Constraints 4](#_Toc309331148)

[1.4 Audience 4](#_Toc309331149)

[1.5 Issues List 4](#_Toc309331150)

[2 Logical Architecture 4](#_Toc309331151)

[2.1 Application Context 4](#_Toc309331152)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Revision |
| 17/11/11 | 0.0 | Niv Ben-David | Started working on the first draft. |
| 17/11/11 | 0.1 | Niv Ben-David | Added first body texts to most of the sections. |
| 19/12/11 | 0.2 | Niv Ben-David | Minor tweeks around the file, removed partsthat were out of the scope of the top-leveldesign document. |
| 19/12/2011 | 0.3 | Michael Dimenshtein | Added Logical Architecture |

1. Introduction
	1. Overview

*Hitch!* is a socially-based transportation solution devised to connect between consumers of (one-time and/or repeating) rides and the providers of such rides such as, but not limited to, private car owners, taxi drivers/stations, car-pooling companies, etc.

Unlike other hitchhiking projects abundant on the net, Hitch! gives the user a large suite of different ways to get from point A to point B, everything in order to provide the user with the product which they require.

* 1. Design Goals and Non-goals
		1. Goals

Hitch! will define a connection between two types of users: a **passenger** and a **driver**, as well as the ‘Currency’ which they trade – a **ride**. A ride can be either one-time or repeating.

The application will support the following features:

1.2.1.1 Create a new user profile.

1.2.1.2 Create a new one-time ride.

1.2.1.3 Request to join an existing ride.

1.2.1.4 Approve/reject a join request to a ride.

1.2.1.5 Search for a ride by a pre-defined set of conditions (time, origin, destination, etc.).

In addition, the application would be able identify a user by their Facebook profile and register them via the data provided by the Facebook application APIs (Open GraphorFQL). Users will be able to see other users with whom they have mutual friends, interests, workplaces, etc.

Users (ride providers as well as consumers) would be able to see each other on a live map and make real-time connections. This feature will enable a passengers to locate a potential nearby ride or locate the driver which whom a ride was scheduled.

* + 1. Non-goals

The application may not, at the first stage, support secondary features such as:

1.2.2.1 Deleting an existing user.

1.2.2.2 Deleting an existing ride.

1.2.2.3 Canceling a join request.

Also, the application would not be able to connect to other data providers (like Twitter or Google+), though the option to extend the database to those providers will be considered during the design and implementation processes.

* 1. Dependencies, Assumptions and Design Constraints
		1. Dependencies

In order to connect between the smartphone end-point and the server an internet connection is required. The application may also depend on a location provider (such as a GPS) system in the smartphone and 3rd -party map providers (such as the stock map application of the Windows Phone or Google Maps).

As a consumer of social network data, the application will depend on the data provided by the Facebook developer APIs (Open Graph or FQL) and the OAuth 2.0 protocol access that data.

* + 1. Assumptions

Since the primary users of the application in its current state can be safely assumed to be only those involved in its production and members of the Tel-Aviv University's Department of Computer

Science faculty, input from the user can be assumed to be safe and not intended to attack or harm the application or server. According to this, only basic safety checks will be performed on user input on the site or application.

* + 1. Design Constraints

Though the application will support a requested/proposed fee field for rides, the exchange system itself will not be implemented and will be assumed to work without interference from the application.

Also, as the application cannot control users’ behavior, users will be assumed to behave accordingly, i.e. pay their passengers fee, arrive to their meeting points (either as drivers or passengers) on time, etc. No functionality to "report" misbehaving users will be implemented.

* 1. Audience

The application is intended for users who are seeking either one-time rides to some destination or a repeating ride to a usual location as part of their routine (work, etc.) as well as for drivers who wish to add passengers to rides they already planned (either as their routine or occasionally) that will accompany them and share the ride’s expenses, or in the case of professional drivers such as bus or cab drivers, pay the ride's fee.

* 1. Issues List

There are no fundamental issues foreseen at the current stage.

1. Logical Architecture
	1. Application Context

Azure

3rd Party Map Application

(Google Maps)

Phone GPS

Hitch! Application

Hitch! Server

Facebook API

Web Service