Transort4You
Design Description

Version 2.0

Revision History

Date	Version	Description	Author
2010-10-8	1.0	Initial Draft	Toni Pivcevic
2010-10-14	1.1	Added TUA Architecture description	Dino Bartosak
2010-01-16	2.0	Addition of sections 4.1.2,4.1.3,4.2	Gaurav Kushwaha

Table of Contents

1. INTRODUCTION 5

1.1 PURPOSE OF THIS DOCUMENT 5
1.2 INTENDED AUDIENCE 5
1.3 SCOPE 5
1.4 DEFINITIONS AND ACRONYMS 5
1.4.1 Definitions 5
1.4.2 Acronyms and abbreviations 5
1.5 REFERENCES 5

2. EXTERNAL INTERFACES 6

2.1 TRANSPORT UNIT APPLICATION EXTERNAL INTERFACES 6
2.2 TRANSPORT MAIN APPLICATION EXTERNAL INTERFACES 6
2.3 TRANSPORT WEB APPLICATION EXTERNAL INTERFACES 6

3. SOFTWARE ARCHITECTURE 6

3.1 CONCEPTUAL DESIGN 6
3.2 SYSTEM SPECIFICATION 6
3.3 ERROR HANDLING 7

4. DETAILED SOFTWARE DESIGN 7

4.1 SYSTEM ARCHITECTURE 7
4.1.1 Transport Unit Application Architecture 8
4.1.2 Transport web application 8
4.1.3 Transport Main Application 9
4.2 TRANSPORT MODEL AND DATABASE 9

5. APPROVALS 11

1. Introduction

1.1 Purpose of this document

This document will identify and describe the overall system design and architecture. All decisions about system architecture have been taken after extensive requirement gathering and analysis, in order to fully abide with the most current functional and non-functional requirements of the project. Detailed study of this document will provide the reader with a general overview of the overall system design.

1.2 Intended Audience

Intended audience includes:

- Team members
- Project leader
- Supervisor(s)
- Customer(s)

1.3 Scope

An overall top-down overview of the system will be presented. The system will be presented to the reader through a component-based description and analysis, since the system itself consists of multiple remotely located components. The system will also be presented through a layered perspective as it also applies.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
Transport line	One line of transport network.
Route section	Journey on a section of transport line performed by user.
Route	Journey performed by user. Sorted collection of route sections.

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
SMS	Short Message Service
Wi-Fi	Wireless Fidelity
Transport4You	Transport For You
TU	Transport Unit
TMA	Transport Main Application
TUA	Transport Unit Application
TWA	Transport Web Application

1.5 References

- Project specification by SCORE

- http://score-contest.org/2011/projects/DiNittoRossi.Transport4You.pdf
- Project homepage
  - http://www.fer.hr/rasip/dsd/projects/transport4you1

2. External interfaces

2.1 Transport Unit Application external interfaces

Interface name	Description	Direction
GPRS device	TUA has to send/receive GPRS data.	TUA→GPRS device
WIFI device	TUA has to detect MAC addresses of all users currently in the TU's WIFI network range.	TUA→WIFI device
Bluetooth device	TUA has to detect MAC addresses of all users currently in the TU's Bluetooth network range.	TUA→Bluetooth device
Event Handler	TUA must handle events from TU: • TU arrives at station (i.e. doors open) • TU leaves station (i.e. doors close)	TU event fire→TUA
GPS device	TUA has to know TU's GPS coordinates.	TUA →GPS device

2.2 Transport Main Application external interfaces

Interface name	Description	Direction
SMS gateway	TMA has to send SMS message notifications to users.	TMA → SMS gateway

2.3 Transport Web Application external interfaces

Interface name	Description	Direction
Credit-card payment service	TMA has to provide payment by credit-card.	TMA → credit card service

* Direction indicates which component is hidden behind an interface.

3. Software architecture

3.1 Conceptual design

The system is comprised of multiple components that interact with each other in various ways to fulfill the functional and non-functional requirements. These components are:

- Transport Main Application
- Transport Unit Application
- Transport Web Application

There is only one instance of the Transport Main Application running on a mainframe server. There is also only one instance of the Transport Web Application (for ease of deployment, it can be running on the same mainframes server). There are multiple instances of the Transport Unit Application and each instance is running on a client machine inside a public transportation Transport Unit travelling around the city. Interactions and communication protocols between these components is described in latter sections.

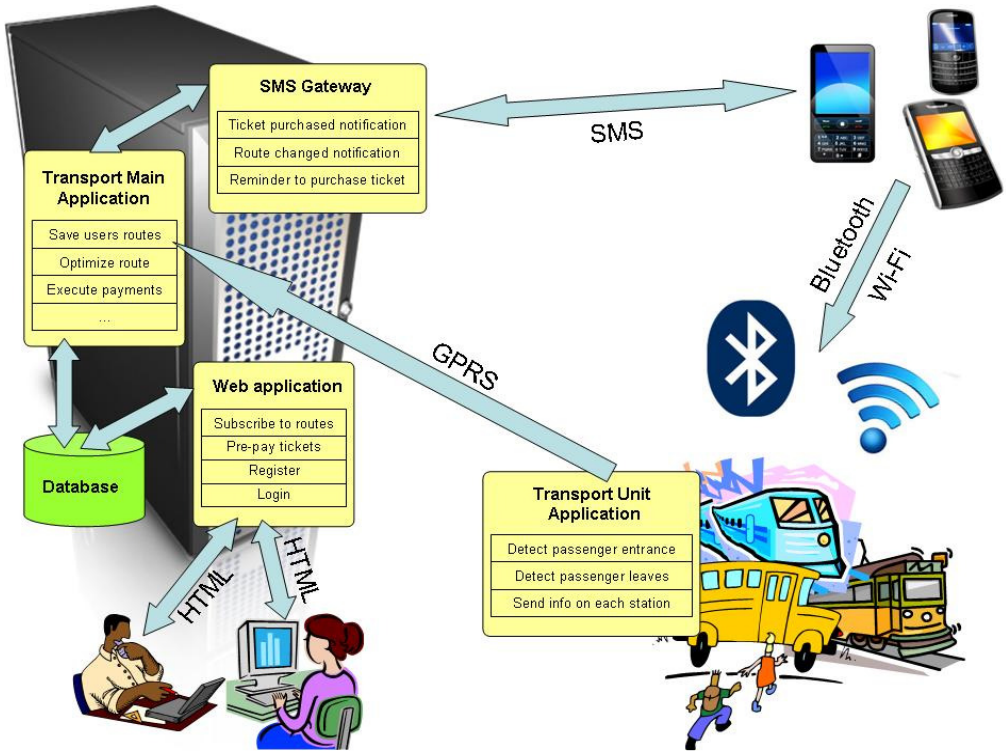
3.2 System specification

3.3 Error handling

Error	Action
TUA failure	Restart, redetect users inside TU, send info to TMA if appropriate.
TMA failure	Restart, notify users/ticket control about situation.

4. Detailed software design

4.1 System architecture



Picture 4-1 System Architecture

Typical interaction and communication between remote components is show in Picture 4-1. Basic interaction between the components are:

Transport Web Application:

- User registration, ticket purchase, and route management is persisted to the database.

Transport Unit Application:

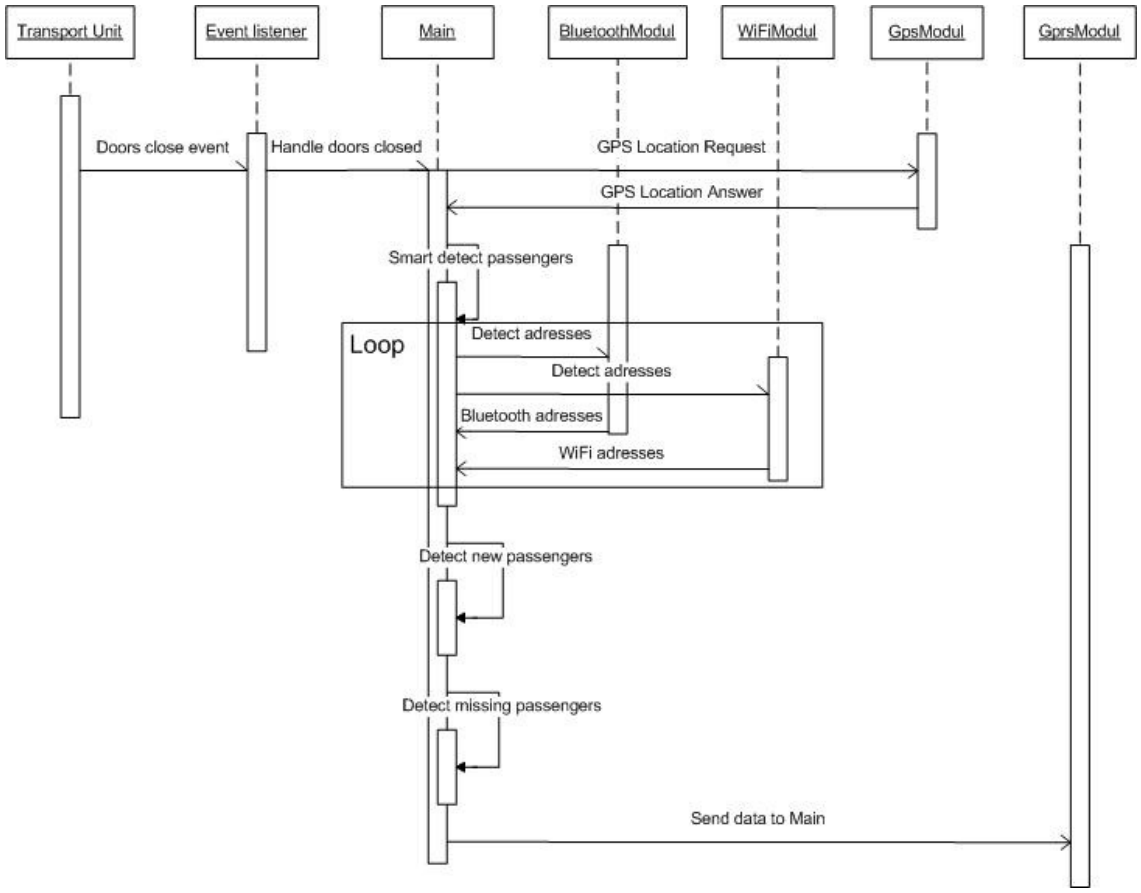
- Uses the WIFI or/and Bluetooth device to smartly detect users inside the TU.
- Uses the GPS device to find out TU's GPS coordinates.
- Uses the TU event system to find out when the TU arrives/leaves a station.
- Uses the GPRS device to send user information to the TMA after leaving each station.

Transport Main Application:

- Uses the GPRS device to receive user information from multiple TUA (each TU leaving a station in the city).
- Uses the database to identify the users by their WIFI/Bluetooth MAC address.
- Uses the credit card system to perform ticket purchase (if necessary).
- Uses the database to identify users habitual routes.
- Uses SMS gateway to send notifications (if necessary).

4.1.1 Transport Unit Application Architecture

TUA uses TU event system to find out when the TU arrives/leaves each station and than smartly detects passengers inside TU.



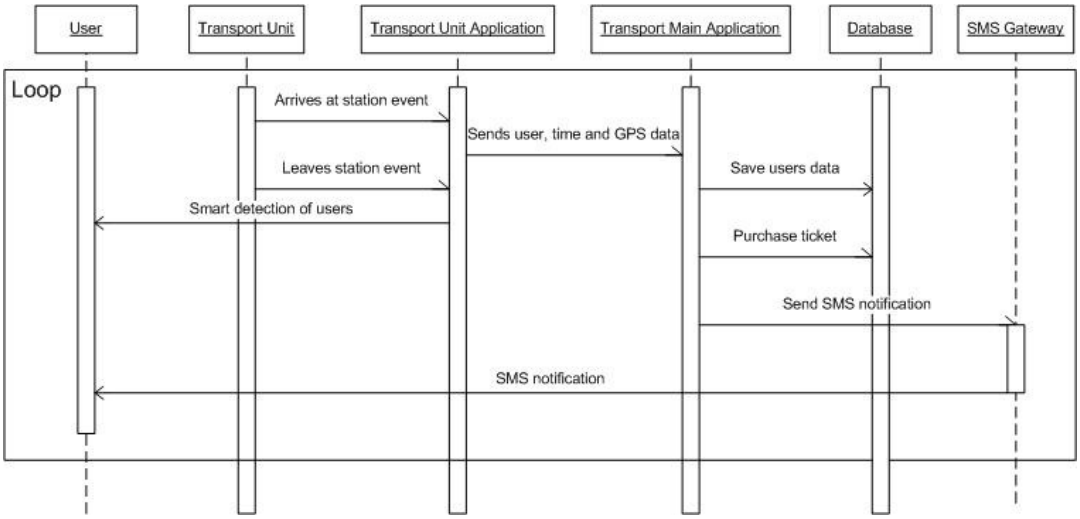
Picture 4-2 Transport Unit Application Sequence Diagram

When the TU triggers the event that TU leaves the station (doors closed), like in Picture 4-2, TUA smartly detects which passengers are actually in the TU. TUA identifies passengers that have entered TU and passengers that have exited TU from the last station. This information together with the GPS location and the current time is sent to the TMA. Smart detection is important so there would be no misinformation. E.g. if some user is in TU's Bluetooth/Wi-Fi range but not inside TU, he should not be affected.

4.1.2 Transport web application

The Transport Web Application (TWA) is Transport4U web interface. Main purpose of TWA is to offer users easy registration into the system. TWA is also used for presentation of news about transport network problems or changes. The administrative section of the application offers easy input of problems and changes to the

network system. TWA and TMA are connected to the same persistence layer, making any additional coupling between the two applications unnecessary.
Picture 4-3 shows the basic flow and interactions between components of the Transport4U system, during a user's journey. More advanced features, such as standard route identification or line interruption notification is not shown on this diagram.



Picture 4-3 – Sequence diagram for automatic detection and billing of a user

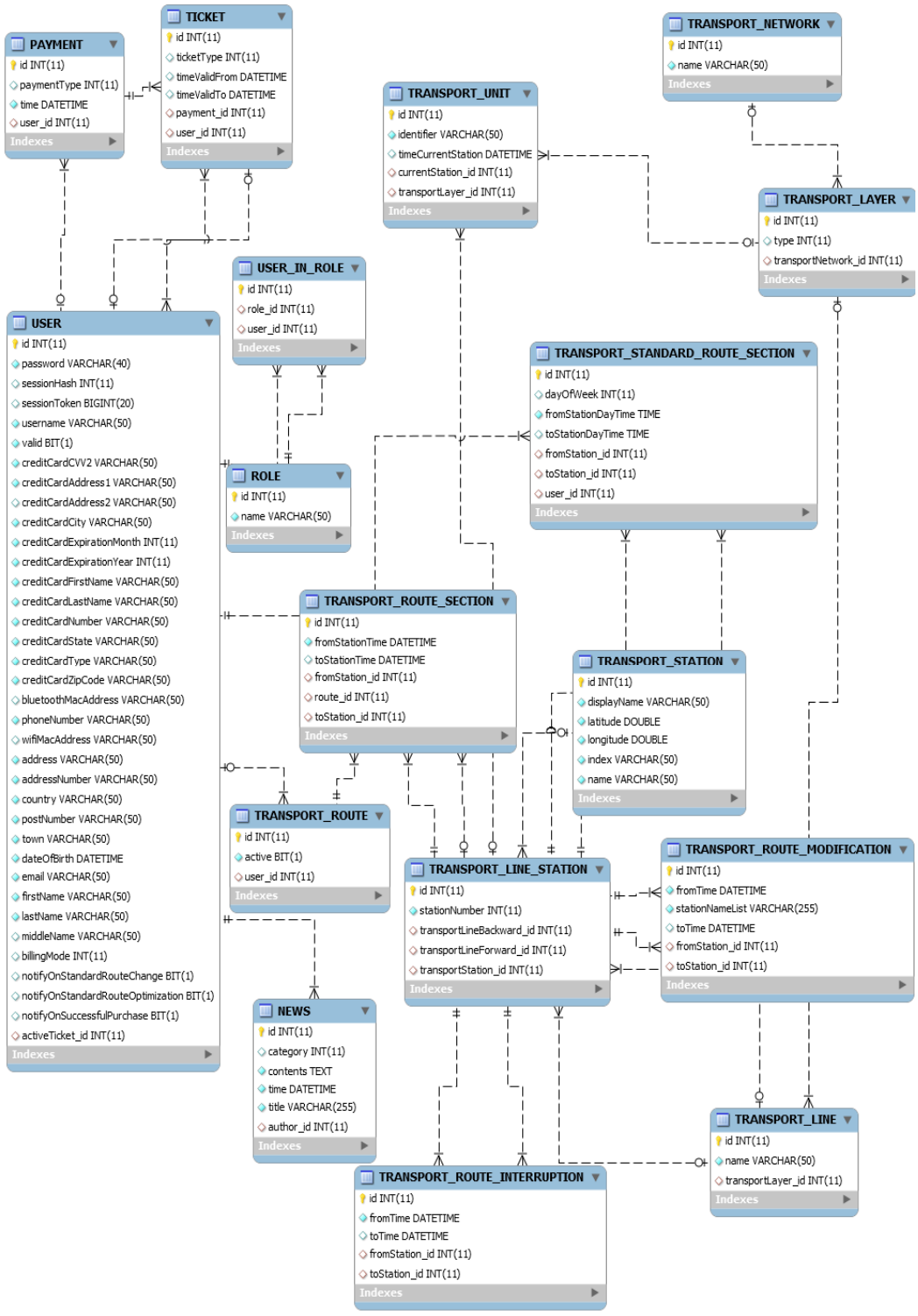
4.1.3 Transport Main Application

The Transport Main Application (TMA) is the mainframe of the Transport4U system. It contains the behavior of the system and it is composed of several sub-modules a) notification module, b) standard route identification module, c) user data handler module, d) billing handler module, e) payment module, f) route optimization module. All these modules represent general behavior of the Transport4U system and new implementations can be integrated, with the existing system, very easily. TMA is the server of the system and only one instance of this application is deployed.

4.2 Transport Model and Database

The Transport model contains mainly domain related classes that require persistence to a relational database. The persistence layer was implemented using an ORM (object-relational mapper) specifically JPA (Java Persistence API), using Hibernate as the underlying provider. As the underlying database we used MySQL, but the mapping is defined in such a way that changing to a different database is fast and easy. The database schema (which is viewable in Picture 4-4) is automatically generated by the defined class structure. Apache Ant[1] build scripts were used for database creation and loading of initial data.

Database access is implemented using the DAO (Database Access Object) design pattern. All queries towards the database are hidden behind interfaces. The DAO design pattern approach provided us with flexibility to change the application's persistence mechanism over time without the need to re-engineer application logic that interacts with the persistence layer.



Picture 4-4 ER Diagram

5. Approvals

Name	Title	Date yyyy-mm-dd	Signature

[1] Apache Ant: (http://ant.apache.org/) - Apache Ant is a Java library and command-line tool who's mission is to drive processes described in build files as targets and extension points dependent upon each other.