

---

## Design Document for «Almaty Today» project

---

Author: Group 5

Erbol Arynbeke

Slamkul Beknur

Zhiyenbay Symbat

Ziyat Bekbol

Version	Date	Author	Change
0.1	12.03.2016	SM	Initial Document

---

## Table of Contents

---

<b>1. Introduction.....</b>	<b>3</b>
1.1. Purpose .....	3
1.2. Scope .....	3
1.3. Definitions, Acronyms, Abbreviations .....	3
1.4. Design Goals .....	3
<b>2. References.....</b>	<b>4</b>
<b>3 Decomposition Description .....</b>	<b>5</b>
<b>4     Dependency Description .....</b>	<b>8</b>
<b>5     Interface Description .....</b>	<b>9</b>

## 1. Introduction

### 1.1. PURPOSE

In this document, written description of the mobile application «Almaty Today».

### 1.2. SCOPE

Our program is mobile, it is associated with the server and a database.

### 1.3. DEFINITIONS, ACRONYMS, ABBREVIATIONS

Term	Description
Client	The man who will be using our software.
Server	All data that are required by the user are stored here.
Database	All data is stored here.
Parsing algorithm	User requirement information from the database.
News	Enters all the latest news from the database.
Send a message	The user can send a message.

### 1.4. DESIGN GOALS

1. Reliability. All the data on the database, and the server must be maintained and reliable.
2. Maintainability. If an error occurs on the server, it automatically has to report it and to repair.
3. Extensibility. Anyone who has a smart phone and the internet will be able to use our program.
4. Response Time. These should be immediately displayed to the user when he asks for news.

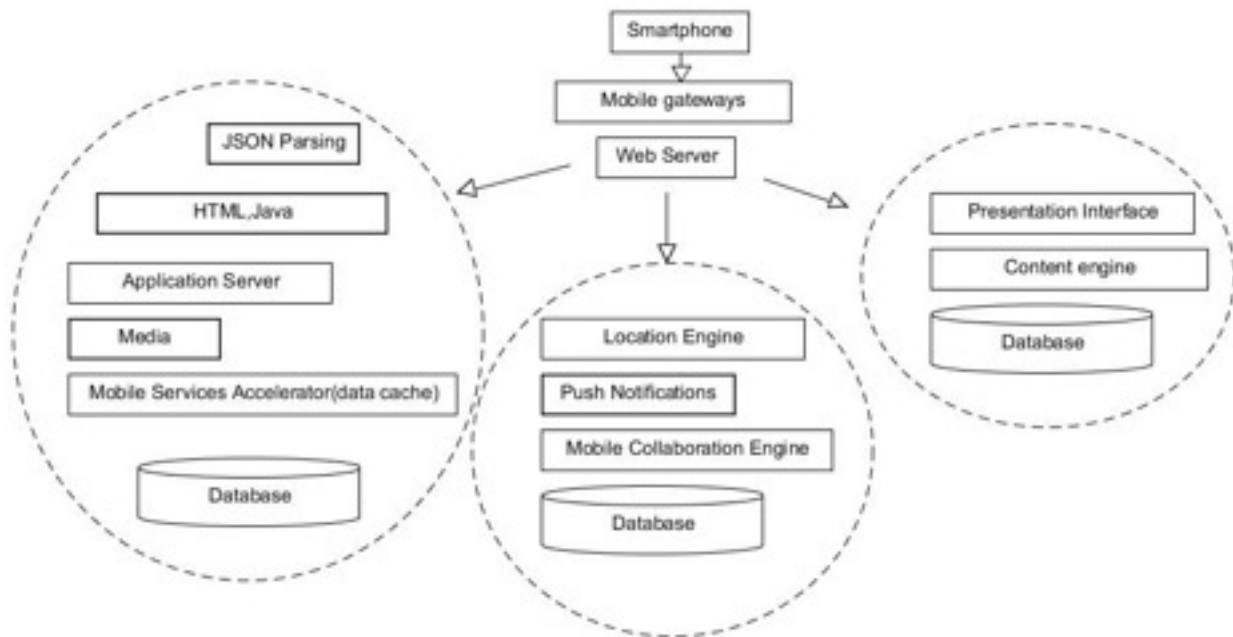
## 2. References

(If any)

### 3 Decomposition Description

#### 3.1 MODULE DECOMPOSITION

Architectural diagram



##### 3.1.1 Internet Users (Smartphone) Description.

Anyone who has an Internet connection on your phone. You can take any news that are on the server and on the data meringue.

##### 3.1.2 Web Server Description.

All data will be placed not only in the mobile applications and online Almaty Today. There, the user can find the information he was looking for or that it became interesting.

##### 3.1.3 JSON Parsing Description.

With this application we can all the data that shows on the site, show the same on a mobile application.

#### **3.1.4 Database Description.**

All data is stored in the database. Regarding the request to the server, the database provides data.

### **3.2 CONCURRENT PROCESS**

#### **3.2.1 Internet Users (Smartphone) Description.**

1. When you open the app the user can see the main page.
2. The user selects which wants to see the news.
3. The user can send a message to the administrator.
4. The user can see the weather and get a notification to the account of the latest news.
5. You can buy online tickets for a concert, theater, etc.

#### **3.2.2 Web Server Description.**

All data will be placed not only in the mobile applications and online Almaty Today. There, the user can find the information he was looking for or that it became interesting.

#### **3.2.3 JSON Parsing Description.**

With this application we can all the data that shows on the site, show the same on a mobile application.

#### **3.2.4 Database Description.**

All data is stored in the database. Regarding the request to the server, the database provides data. Then sent to the server and further to the mobile application.

### **3.3 DATA DECOMPOSITION**

3.3.1 <Class 1> Description

3.3.2 <Class 2> Description

### **3.4 STATES**

3.4.1 <State/System 1 > Description

3.4.2 <State/System 2> Description

## **4 Dependency Description**

### **4.1 INTERMODULE DEPENDENCIES**

### **4.2 INTERPROCESS DEPENDENCIES**

### **4.3 DATA DEPENDENCIES**



## 5 Interface Description

### 5.1 MODULE INTERFACE

#### 5.1.1 Android studio code interface

##### 5.1.1.1 Interface between Android studio code and Database connection library

1) public function createUser(var id:Integer,var name:String, var surname:String):Boolean;

is used to create new user in database,

Parameters:

id = the unique id of user, it will be given automatically unique number,

name = the name of the user,

surname = surname of the user,.

Return type: Boolean, in case of error returns false. 2)public function deleteUser(var id:Integer):

Boolean; is used to delete the user from database,

Parameters:

id = the unique id of each user,

Return type: Boolean, in case of error returns false.

3)public function getGeneralInfo(var id:Integer):ResultSet; is used to get all important user's information from database, Parameters:

id = the unique id of each user,

Return Type: ResultSet – result from given query,

4)public function getConditions(var id:Integer) :ResultSet;

is used to get user's all conditions (like illness history) from database Parameters:

id = the unique id of each user,

Return Type: ResultSet – result from given query,

5)public function setConditions(var id:Integer, var date:String, var status: String, var str: String ): Boolean;

is used to set user's all conditions (like illness history) to database

Parameters:

id = the unique id of each user,

date = the date of illness period,

status = the status of condition,

str = the condition (all about illness)

Return type: Boolean, in case of error returns false.

6)public function getReceipts(var id:Integer) :ResultSet;

is used to get user's Receipts (list of drugs, which needs user) Parameters:

id = the unique id of each user,

Return Type: ResultSet – result from given query,

7)public function setReciepts(var id:Integer): Boolean;

is used to set user's Reciepts(list od drugs, which will need user) Parameters:

id = the unique id of each user,

Return type: Boolean, in case of error returns false.

8)public function getAllergies(var id:Integer) :ResultSet;

is used to get user's Allergies(list od drugs or procedures which acts negatively to user) from database

Parameters:

id = the unique id of each user,

Return Type: ResultSet – result from given query,

9)public function setAllergies(var id:Integer, var type: String, var des: String): Boolean;

is used to set user's Allergies(list od drugs or procedures which acts negatively to user) to database

Parameters:

id = the unique id of each user,

type = the name of allergy or drug,

des = the description of allergy, how it acts to user

Return type: Boolean, in case of error returns false.

10)public function getProcedures(var id:Integer) :ResultSet; is used to get user's

Procedures(medical treatment) Parameters:

id = the unique id of each user,

Return Type: ResultSet – result from given query,

11)public function setProcedures(var id:Integer, date: String, str: String): Boolean; is used to set user's Procedures(medical treatment) to database

Parameters:

id = the unique id of each user,

date = date of illness period,

str = the condition (the procedure)

status = it will be gives automatically (present) Return type: Boolean, in case of error returns false.

12)public function getTestResults(var id:Integer) :ResultSet; is used to get user's test results(any medical tests) Parameters:

id = the unique id of each user,

Return Type: ResultSet – result from given query,

13)public function setTestResults(var id:Integer, date:String, type :String, res: String): Boolean;  
is used to set user's test results(any medical tests)

Parameters:

id = the unique id of each user, date = the date of tests,  
type = the type of any tests(IELTS), res = the result of test.

Return type: Boolean, in case of error returns false.

14)public function getImmunization(var id:Integer, date:String, type:String): ResultSet;  
is used to set user's Immunizations from database,

Parameters:

id = the unique id of each user,  
date = the date of immunization process,  
type = the type of immunizations,

Return Type: ResultSet – result from given query,

15)public function getNotices(var id:Integer) :ResultSet; is used to get user's Notices from  
database,

Parameters:

id = the unique id of each user.

Return Type: ResultSet – result from given query,

17)public function setNotices(var id:Integer, date: String, str: String) : Boolean; is used to get user's  
Notices (Analyze of illness and hospital or doctor name)from database,

Parameters:

id = the unique id of each user,  
date = the date of notice(when it given),  
str = the description (maybe illness type, the hospital which responsible this user) Return type:  
Boolean, in case of error returns false.

### **5.1.1.2 Interface between Android studio code and Server code**

5.1.1.2.1 public databaseListener();

In Android studio we call constructors of server class which extends from thread  
It listens for database changes , in case if changed call panelUpdater() Method

Parameters: None

Return Type: None

Return type:String – Name of panel which should be updated

5.1.1.2.2 public String panelUpdater();

Finds out which table changed, then sends string which says which panel should be updated

Parameters:None

## **5.1.2 Android studio code interface**

5.1.2.1 Interface between server code and Android studio code a

5.1.2.1.1 See section 5.1.1.2

5.1.2.2 Interface between Android studio code and Database connection library

5.1.2.2.1 public String checkDatabase();

This method checks for each table for its change

Parameters: None

Return type: String – returns name of table that was updated, returned string is null if nothing changed

## **5.1.3 Database interface**

5.1.3.1 See sections 5.1.1.1 and 5.1.2.2

## **5.2 PROCESS INTERFACE**

### **5.2.1 Android studio Main process**

Description: This process shows all graphical interface of the system

5.2.1.1 Process is created when the application started

5.2.1.2 Terminated when applications close button is pressed

5.2.1.3 All other threads will be killed if this main thread stops

### **5.2.2 Database Listener process**

5.2.2.1 This thread is created after Main process acquires all information from Database

5.2.2.2 Database listener process interacts with panel Server process

5.2.2.3 This process will be terminated automatically if Main thread of process is killed

### **5.2.3 Server process**

5.2.2.1 This thread is created after Database Listener process

5.2.2.2 Mainly this thread interacts between Main thread and Database listener tread

5.2.2.3 Thread is terminated when one of other two threads terminates