Notifive Calendar Sync

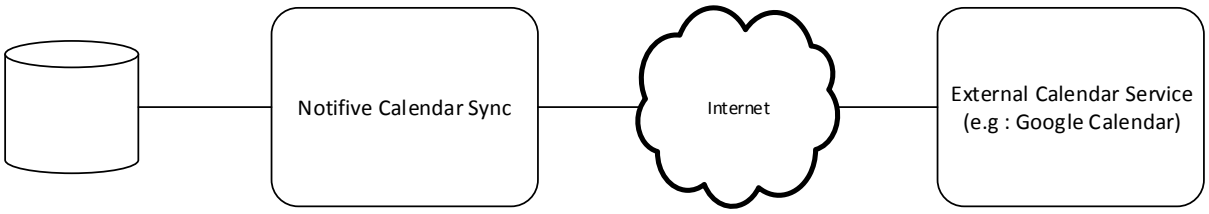
Revision Info

V0.1	General Info, Technical Requirements and Milestone 1	March 11, 2014

1 Introduction

The purpose of this software program are the following

1. Develop a web application called “Notifive Calendar Sync” using Java, hence forth called as NCS
2. NCS should read appointment data from its database table and populate the appointment data on an external calendar service like Google Calendar
3. NCS should read appointment data from an external calendar service like Google Calendar and NCS database Application

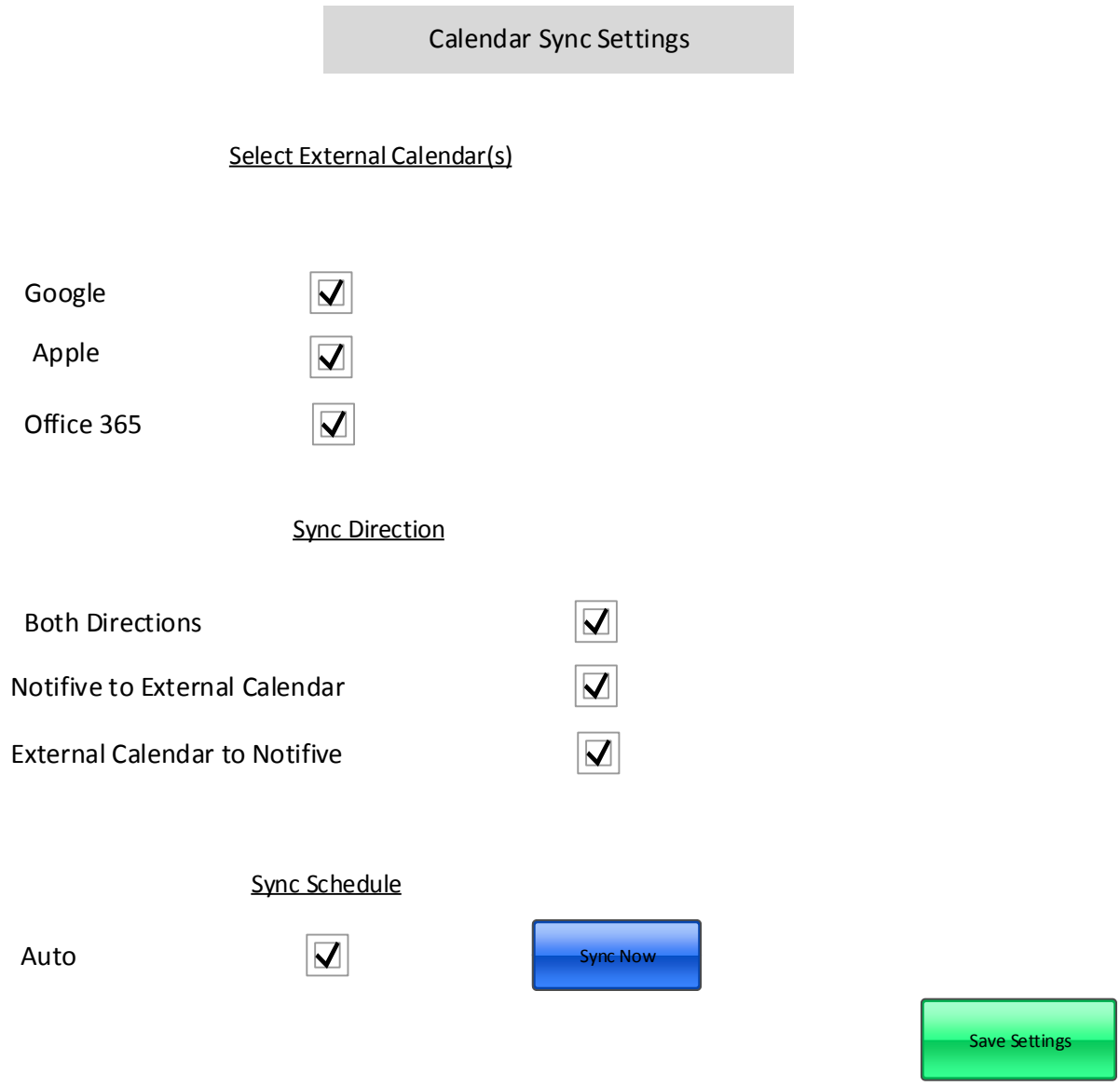


2 Technical Specifications

1. NCS should use Java 6 or 7
2. NCS should support Tomcat as the application server
3. NCS should support Postgres 9.x with JDBC connection
4. NCS should support OAuth 2.0 for authentication with external servers
5. NSC should NOT store any user name or passwords in databases or in source code
6. All configuration parameters should be configurable via database or web.xml file
7. NCS should support log4j for logging

3 Calendar Sync Web page

Mock up of how the Calendar Sync page should look like is provided below.



NOTE: You will be provided a sample page and CSS to use

NOTE: This page will be repeated for each Notifive Calendar, selectable by dropdown menu or tab

4 How it works

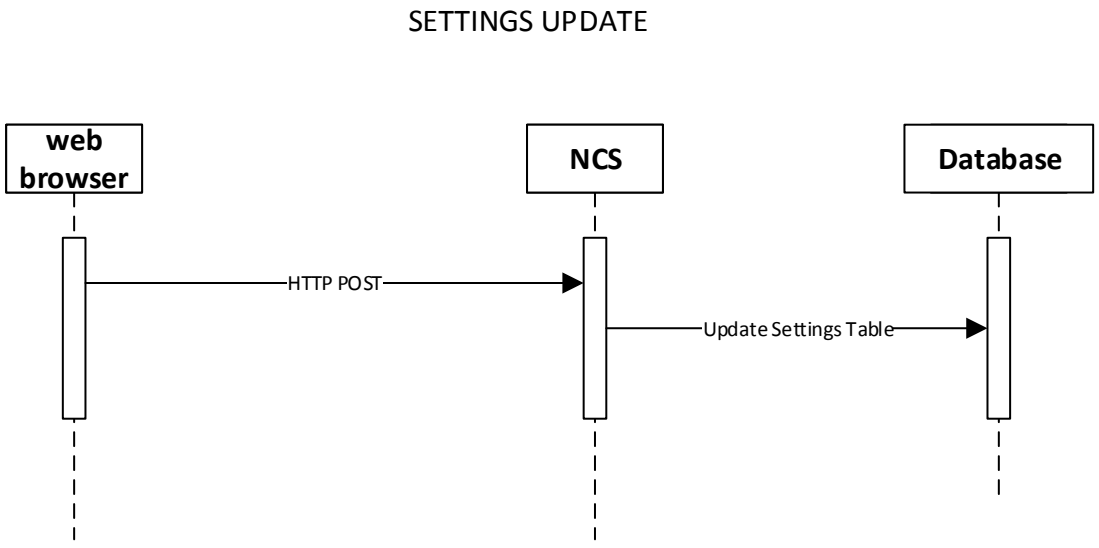
Notifive appointment data will be inserted into the appointment table whenever the user set up an appointment. Depending on the settings, the appointment data will be 'pushed' to the external server.

Appointment data will be pushed to the external calendar either when the data is inserted into the database or when the user clicks the 'Sync Now' button

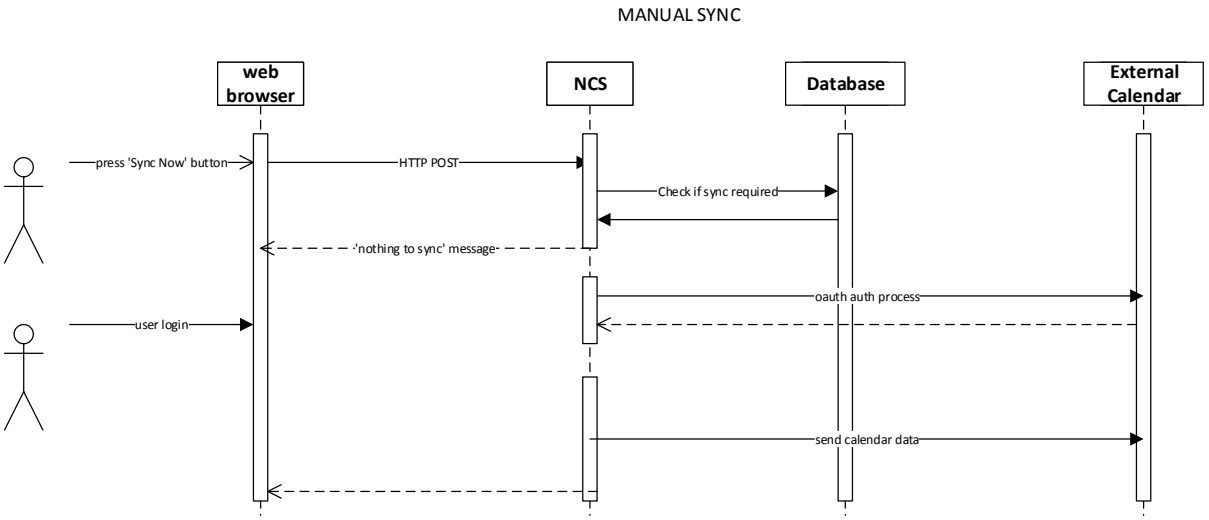
Depending the settings, appointment data from external server will be pulled from the external server and gets inserted into the Notifive database table.

5 Sequence Diagram

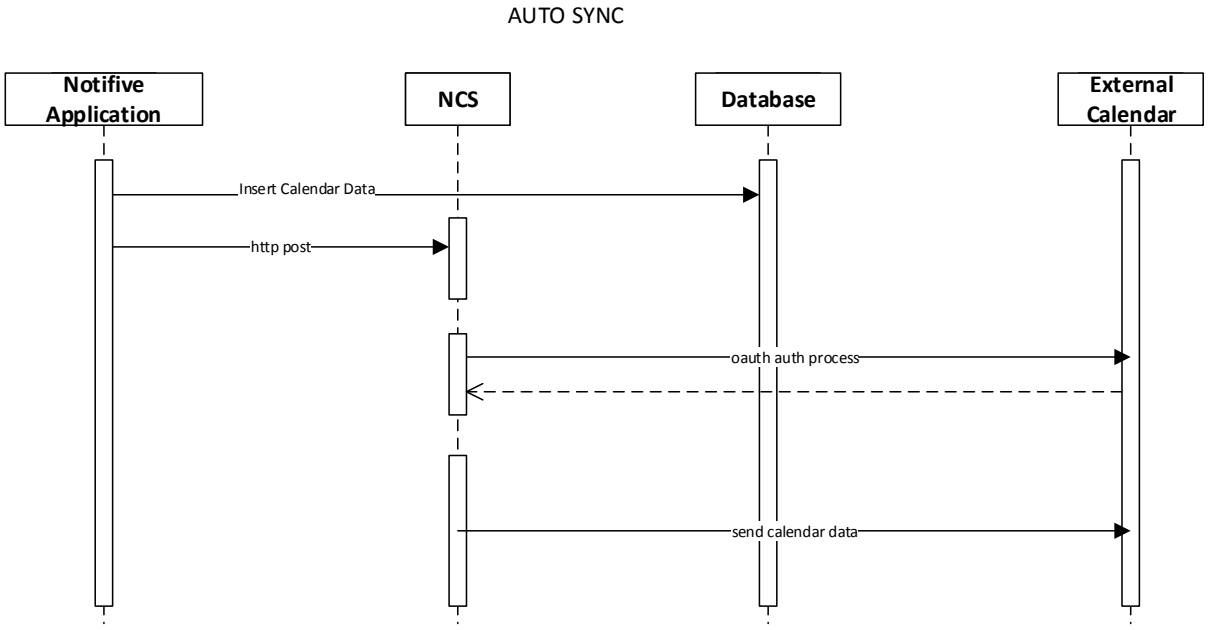
5.1 User sets / updates the settings



5.2 Manual Sync



5.3 Auto Sync



6 Table Schema

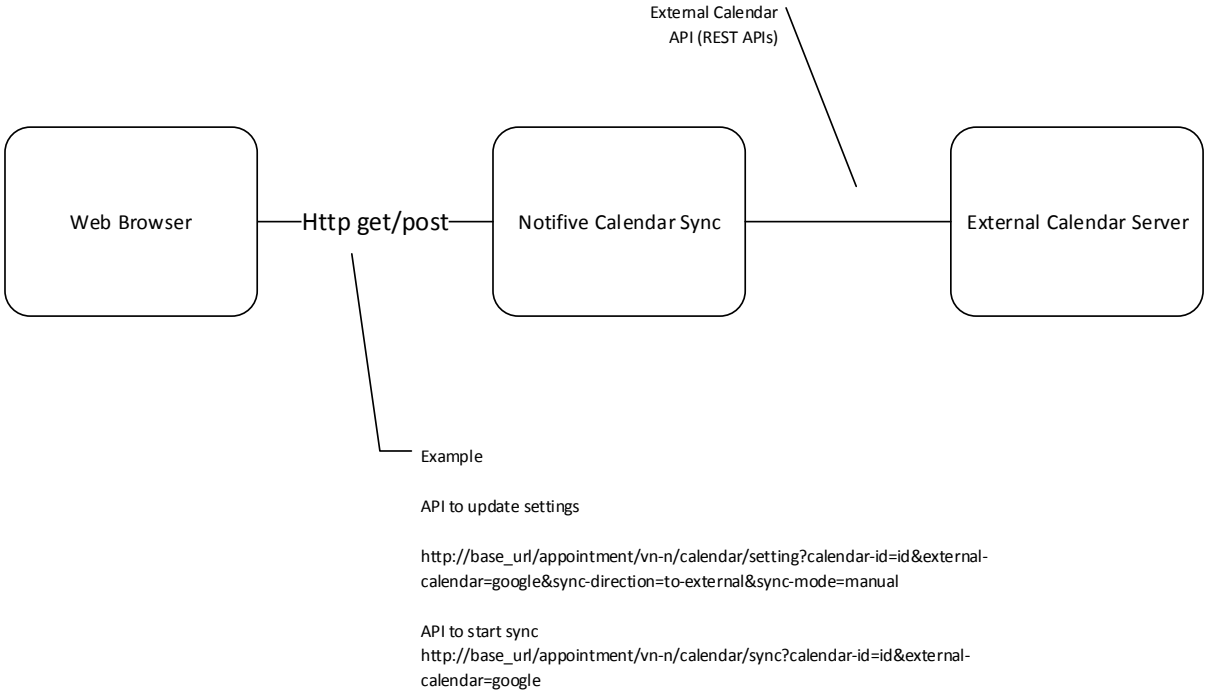
table_appointment

```
CREATE TABLE table_appointment
(
appointment_id serial NOT NULL,
transaction_id int,
client_id character varying NOT NULL,
application_id character varying NOT NULL,
title character varying,
name character varying,
start_date timestamp with time zone,
end_date timestamp with time zone,
duration int,
all_day boolean,
mobile character varying,
email character varying,
confirmation_status character varying,
location character varying,
notes character varying,
sync_status character varying,
calendar_id int
) WITH (
OIDS=FALSE
);
ALTER TABLE table_appointment
OWNER TO postgres;
```

table_calendar

```
CREATE TABLE table_calendar
(
calendar_id serial NOT NULL,
client_id character varying NOT NULL,
application_id character varying NOT NULL,
name character varying NOT NULL,
start_time character varying,
end_time character varying,
sync_external_calendar_type character varying,
sync_direction character varying,
sync_method character varying
) WITH (
OIDS=FALSE
);
ALTER TABLE table_calendar
OWNER TO postgres;
```

7 API



7.1 Settings

HTTP Method – GET or POST

Endpoint / URL - http://base_url/appointment/vn-n/calendar/setting

Query String Parameters

Parameters	Value	Description
calendar-id	Integer	Id of the notifive calendar this settings applies
external-calendar	String (Google, Apple,Office365)	Name of the external calendar system
sync-direction	String (both,notifive-to-external,external-to-notifive)	Defines how the direction of the synchronization
sync-mode	Auto, Manual	Defines the mode of the sync

Example

http://base_url/appointment/vn-n/calendar/setting?calendar-id=id&external-calendar=google&sync-direction=to-external&sync-mode=manual

7.2 Sync

HTTP Method – GET or POST

Endpoint / URL - http://base_url/appointment/vn-n/calendar/sync

Query String Parameters

Parameters	Value	Description
calendar-id	Integer	Id of the notifive calendar this settings applies
external-calendar	String (Google, Apple,Office365)	Name of the external calendar system

Example

http://base_url/appointment/vn-n/calendar/sync?calendar-id=id&external-calendar=google

7.3 Additional Note:

You may use ***XMLHttpRequest*** or Ajax libraries which provides this functionality.

8 Testing

Tests with external servers can be done using test accounts.

Appointment database table can be populated with appropriate data for testing.

9 Milestones

9.1 Milestone 1

- Support Google Calendar only
- Support one way synchronisation only (from Notifive to Google)
- Support log4j
- Support Setting page
- No JSP code. Only static HTML +CSS + JS.
- All configurable parameters should be in web.xml or database tables (config table)
- Only manual sync required (using ‘sync now’ button)

9.2 Milestone 2

Will be provided in v0.2 after milestone 1 completion

9.3 Milestone 3

Will be provided in v0.3 after milestone 2 completion