Dept of Computer Science and Information Systems

BSc Final Year Project Form (2015/2016)

1. Proposal

The student should complete parts 1(a), 1(b) and 1(c) below, and then agree the maximum pocket values with the supervisor and put these in part 2(a) below. An electronic version of this form should be uploaded to the Final Year Project page on Moodle no later than **Monday 2nd November 2015**.

(a) Student details

Name:	**Module *delete as appropriate***
	Project Type 3 (BUCI026S6)
Email:	

(b) Project details

Title: Automated Notification System using API first design

Objectives:

Create a notification system that is able to send a notice via email or pushes the information to a webhook / API.

Create a API to enable the setup of users and accounts

Create a frontend for the setup

Create a reporting tool for users.

Title: Automated Notification System using API first design

Description:

This project is done as an early proof of concept in the learning and future product design of the Expedia Affiliate Network (EAN).

EAN provides a comprehensive hotel search and booking API for affiliates, which allows the marketing of hotel content.

Affiliates can integrate the EAN API to search, get information and book hotel rooms.

A common issue is that affiliates also would like to get a notification if a booking is cancelled.

The only solution at the moment is for every affiliate to send queries to the EAN API to check the booking status of each booking made and determine changes to the bookings, a process that is difficult as booking data migrates from a short-term (fast) database to a slower long-term system which needs to be managed by the affiliate.

The only current notification capability is an email that is sent to a single email address.

Affiliate developers repeatedly quote 1-2 weeks development time for an API system that can provide the functionality needed to manage cancelled bookings. This causes delays in a successful rollout and can increase the business investment for the whole integration project so it is no longer profitable enough.

This process is also asking different development teams of affiliates to invent similar code, which needs to be tested and approved by EAN tech teams prior to an approved rollout.

Other hotel content providers already have push capability that affiliates can configure and set up.

A better method would be for EAN to handle the logic and send a push notification to an existing affiliate endpoint. As most affiliates are already working with content providers, the EAN push mechanism needs to be configurable in a way that it can fit into existing endpoint solutions.

While individual affiliates could use hardcoded values or simple configuration files to hold their client identification, the proposed prototype will need to hold multiple customer identifiers and the ability to generate a valid signature to authenticate that the system is authorised to make these requests on behalf of the affiliate.

Time permitting on the project, the additional analytics part can identify properties that cancel often, which then can be excluded in sales, leading to a higher customer satisfaction.

Title: Automated Notification System using API first design

Method:

The project will have 3 main components:

The API Data System, which will run in intervals on a server and query existing bookings on the EAN API, writing them into a database and comparing the status of bookings to the previously retrieved status. If the status changes, a notification is send to the affiliate.

The Setup and Lookup API, build to write customer information into the database and retrieve data from the system.

Once the API has been built, a User Frontend is generated to allow entering and retrieving data from account managers. The frontend part can be extended for additional reporting capabilities.

Work plan:

Modular Design of the Project:

- **API Data System: This is the Core part of the project, as it will allow and provide the push functionality and prove that data can be retrieved via the EAN API and pushed forward to an affiliate.**
- **Project native API to allow setting up accounts and retrieving data for reporting.**
- **User Interface to allow setting up accounts using the API**
- **Extension: Build in more reporting logic in the API**
- **Extension: Build reporting pages in the user frontend.**

Due to the modular design approach it is easy to extend certain parts as they become necessary or scale back on other parts if there are time constraints in the development.

College equipment required:

None. The solution will be hosted on a private server.