ITEC809 Project Proposal

Applying Social Networking to the Enterprise – Making an Intelligent Status Tracking Application

Sidney Shek (41419979)

Supervisor: Dr Rolf Schwitter

21/08/2009

Summary

A common problem in the workplace is determining how best to communicate with one another especially between mobile workers. The aim of this project is to develop a low-cost prototype intelligent status tracking system called 'Can We Talk Now?' (CWTN). CWTN is inspired by principles from Social Networking Services (SNS) and makes use of various features from mobile devices (Apple iPhone initially) such as calendars and GPS to automatically populate status information. A rules engine will be used to implement the logic to determine the "best" means of communication between users. CWTN's automated nature should encourage use of the system by busy individuals who would otherwise be unable to maintain status information in standard SNSs. In addition, unlike other systems CWTN aims to be independent of telecommunications providers thereby allowing any enterprise to use it.

The project has been divided into two phases; a feasibility study and technology selection phase conducted last semester and a development phase for the construction of the prototype to be conducted this semester. The development phase will consist of design followed by two coding iterations to produce a demonstrable system at the end of the semester including core server-side and client-side functionality.

1 Project Description

1.1 Background and Motivation

Team communications are essential in an enterprise environment. However, team members are often busy with meetings or delivery for deadlines and may be mobile. As a result, a common problem faced by many individuals is determining the most effective means of communication with another person at a point in time. For example, an urgent question is probably best answered in-person or via a telephone conversation. However if the target of the question is in an important meeting with a client, he/she may not appreciate an interruption.

As an added dimension to this problem, many users now have smartphones (e.g. Apple iPhone or Blackberry). While these users may not always have access to a computer such as while travelling or between meetings, they do have access to the Internet or the cellular network and hence can be contacted, albeit with potentially limited capabilities. By providing accurate status information about these mobile users, it is possible to facilitate phone conversations or other limited forms of communication, thereby allowing productive use of their time.

There are tools such as emails, calendars and instant messaging that partially address the problem of providing up-to-date status information. However they have several issues such as:

1. Calendars are not often kept up to date, and do not always accurately reflect the duration of the meetings or a person's availability (e.g. a person may not be in a meeting, but may be busy and not interruptible).
2. Instant messaging tools typically only provide basic status information to identify whether someone is busy or idle. Also, it is a common occurrence for users to forget to update their status. In addition, instant messaging tools are typically used only on PCs and not mobile devices due to constant network activity draining limited battery life.
3. Emails and other 'word by mouth' status updates may be easily lost or not distributed.

Social networking services (SNS) such as Facebook[1] and Twitter[2] provide a central location for status information that is manually maintained, yet they have demonstrated that near real-time updates to status information can be achieved. This information could then be used by others to determine the best means of communication. It is the application of this principle to mobile workers in the enterprise that is the focus of this particular project.

1.2 Aims, Significance and Expected Outcomes

1.2.1 Project aims

The main aim of this project is to develop a prototype system named CanWeTalkNow? (CWTN) for recording and querying a person's current status. The key aspects of the system are:

- Automatic population of status information based on various sources available from mobile devices. This includes location information based on Global Positioning System (GPS), calendar entries and network availability.
- Provision of inbuilt logic to determine the 'best' means of communication between two users of the system.
- Low development, deployment and support costs so as to encourage uptake of the prototype in pilot and potentially full deployment scenarios.
- Support integration into an enterprise environment for providing status information to other systems, or potentially obtaining necessary user data from other systems.
- Only make use of end user functionality in mobile devices, thereby removing dependencies on telecommunications providers.

The prototype could serve as the basis of a pilot in an organisation where the effectiveness of this particular application can be measured, possibly resulting in future research work or development into a larger scale solution.

1.2.2 Significance

CWTN serves as a platform for a number of potential uses:

- It is a tool designed to improve communications within mobile teams. This could lead to productivity improvements in organisations.
- Successful uptake of the CWTN prototype with resulting productivity benefits could promote SNS deployments in enterprises, which are typically conservative and require quantitative evidence before deployment state-of-the-art technologies such as SNS. These enterprises could benefit from resulting productivity improvements from SNS usage to share knowledge.
- CWTN can also be a research vehicle or test bed into SNSs, use of mobile devices in the enterprise, integration with various enterprise systems, and application of reasoning to SNS knowledge bases.
- CWTN could also be turned into a niche product for use by many enterprises.

[1] See www.facebook.com

[2] See www.twitter.com

1.2.3 Expected Outcomes

The expected outcomes of this project are:

1) Proof of concept system comprising of the following components:
    a. Server application that stores statuses with a rules/reasoning engine.
    b. Client application allowing entry of a person's status. This may be extended to allow querying of other's statuses if time permits, and an application deployable to specific mobile devices.
2) Report documenting:
    a. Analysis and design of the system including UML class and sequence diagrams etc.
    b. Discussion of issues with the system that were identified during development.
    c. Additional enhancements that may be implemented.

Note that the work conducted during this semester is a continuation from the last subject ITEC808, and hence the report submitted as part of ITEC809 can be considered a supplement to the ITEC808 report.

2 Methodology and Plan

2.1 Approach

This project primarily consists of software development, with an initial investigation to finalise the scope. The project will run over two semesters, covering units ITEC808 and ITEC809. In order to fit into ITEC808/809 unit split constraints, the project will run in two phases:

- Phase 1: Feasibility study and development project scoping. This was the main focus during ITEC808. It was an analysis phase to determine requirements for the system, to select appropriate technologies and to document specific issues that may need to be addressed as part of development and deployment of the system. The scope of development iterations for phase 2 was also determined.
- Phase 2: Software development. This will be the focus during ITEC809. A mix of waterfall and iterative development will be used [Sommerville 2007, Ch 4]. Overall system design would be conducted initially to provide to ensure all requirements are captured so as to minimise rework in later iterations. System design would include sequence and class diagram documentation and necessary design decisions. The design would then feed into development iterations focussing on specific requirements. This iterative approach will allow core components of the project to be completed first for submission, with potential for additional development work to be performed if time permits.

  Four coding iterations were identified as part of phase 1 ('Server Core', 'Mobile Core', 'Mobile Enhancements' and 'Web Interface'). Diagram 1 below shows the relationship between the various system components and the coding iterations. Given the time constraints of ITEC809, only the first two will be completed, which represent the minimal working functionality of the system.

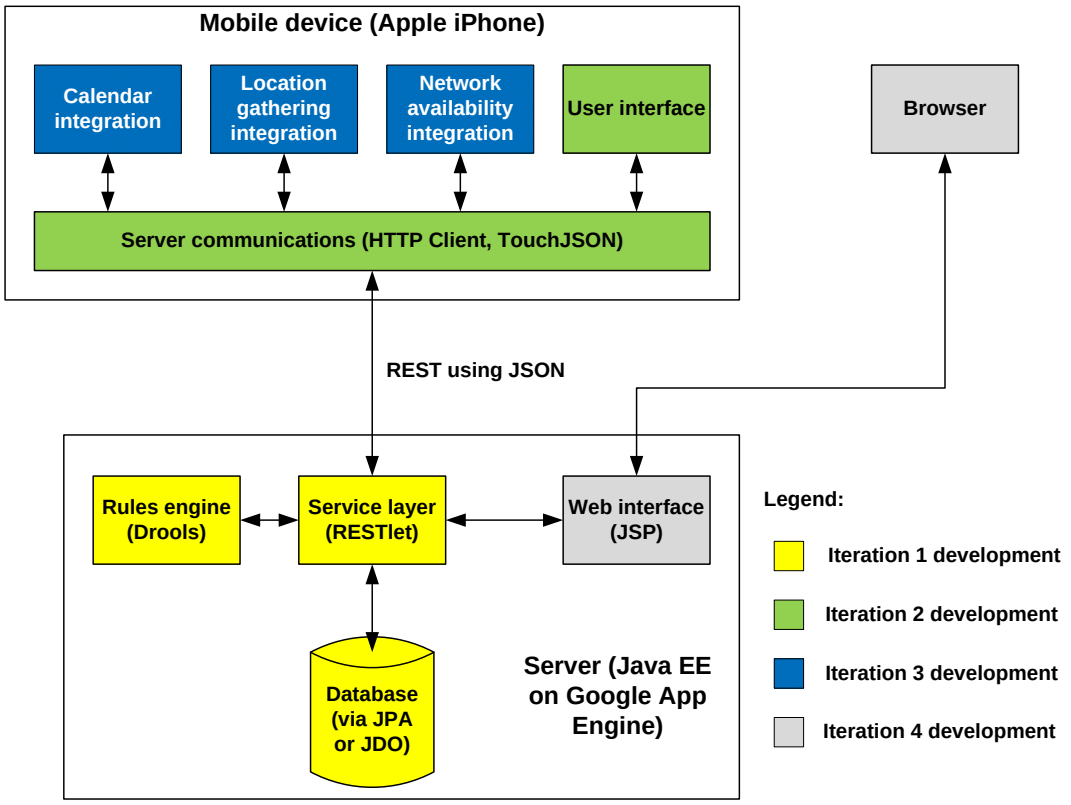

Figure 1 Diagram showing breakdown of high-level system architecture into development iterations

2.2 Task Plan

A detailed Gantt chart covering ITEC809 can be found in Appendix A, which includes effort and delivery dates for tasks and subtasks. The 'work' column for tasks and subtasks shows the estimated effort. 12 hours per week are allocated for the entire duration of the project.

2.2.1 Task 1: System analysis and design [56 hours effort, 16/8/09 – 3/9/09, Weeks 3 – 5]

The purpose of this task is to take the requirements identified during ITEC808 and produce a design for the prototype system to feed into development. It consists of the following sub-tasks which will be used for detailed tracking:

- **1.1. Design server-side data model** [8 hours effort] – This includes producing data model documentation in preparation for developing suitable database schemas and data objects. The documentation will be in the form of UML class diagrams.
- **1.2. Design server-side class model** [8 hours effort] – This includes identifying necessary classes in the server-side code for interfaces, data access objects, and rule engine integration. The documentation will be in the form of UML class diagrams.
- **1.3. Server-side sequence diagram documentation** [8 hours effort] – This includes translating the use of classes identified as part of task 1.2 into UML sequence diagrams.
- **1.4. Design client screens** [8 hours effort] – This includes wireframes or user interface prototypes for client screens specifically for the Apple iPhone. The client screens would include a status querying screen, manual status entry screen (with scope for adding automatic data population) and other screens that are necessary to ensure easy navigation between the two.

- **1.5. Design client-side data model** [8 hours effort] – This includes designing the data model required for the client application with documentation in the form of UML class diagrams.
- **1.6. Design client-side class model** [8 hours effort] – This includes identifying necessary classes in the client application, and documenting the classes using UML class diagrams.
- **1.7. Client-side sequence diagram documentation** [8 hours effort] – This includes documenting the use of classes identified as part of task 1.6 in UML sequence diagrams.
- **System Analysis and Design Complete** – This is the milestone for this task (3/9/09).

Note that this design task will consider requirements for all development iterations as identified during ITEC808. This is considered necessary to minimise rework during development iterations, and also to ensure easy 'pickup' of the design at a later date if the development work were to cease at the end of the semester.

The overall deliverable for this task is completed documentation for system design. This would make up a significant proportion of the project report.

2.2.2 Task 2: Coding Iteration 1 – Server Core [60 hours effort, 3/9/09 – 26/9/09, Weeks 5 – 7]

The purpose of this task is to develop the core server-side components of the system. This includes a data access layer, a service layer for client-server communications, and the rules for determining the 'best' means of communication. It will take the outputs of task 1 'System Analysis and Design', and produce code that can be built and deployed to the target application server (Google App Engine).

The following subtasks will be used for detailed tracking:

- **2.1. Develop data objects** [8 hours effort] – This includes developing Java data objects and database schema based on the server-side data model (from task 1.1)
- **2.2. Develop database access layer** [12 hours effort] – This includes developing database access layer classes identified as part of task 1.2.
- **2.3. Develop REST service** [8 hours effort] – This includes developing service layer classes identified as part of task 1.2 with integration to the database access layer as identified during task 1.3.
- **2.4. Develop rules and rule engine integration** [16 hours effort] – This includes refining rule code from ITEC808 and developing necessary rule engine classes as identified during task 1.2.
- **2.5. Build code for application server** [16 hours effort] – This includes integrating the developed code with the target application server to support data sources and authentication.
- **Coding Iteration 1 Complete** – This is the milestone for this task (28/9/09).

The overall deliverable for this task is completed and operational server-side code for the prototype, made up of individually operational modules from each of the subtasks.

2.2.3 Task 3: Coding Iteration 2 – Mobile Core [64 hours effort, 26/9/09 – 29/10/09, Weeks 9-13]

The purpose of this task is to develop the core client-side components of the system specifically for the Apple iPhone based on the design completed as part of task 1 'System Analysis and Design'. This includes a data access layer, client-server communications code, and basic user interfaces for manual data entry.

The following sub-tasks which will be used for detailed tracking:

- **3.1. Develop data objects** [8 hours effort] – This includes developing data objects and database schema based on the client-side data model (from task 1.5)
- **3.2. Develop data access layer** [16 hours effort] – This includes developing data access layer classes identified as part of task 1.6.
- **3.3. Develop integration to REST service** [16 hours effort] – This includes developing classes identified as part of task 1.6 that are responsible for integrating with the REST service on the server.
- **3.4. Develop status query screen** [12 hours effort] – This includes developing classes and user interface elements for the basic 'status query' screen as designed in task 1.4.
- **3.5. Develop status entry screen** [12 hours effort] – This includes developing classes and user interface elements for the basic manual status entry screen as designed in task 1.4.
- **Coding Iteration 2 Complete** – This is the milestone for this task (26/10/09).

The overall deliverable for this task is completed and operational client-side code for the prototype, made up of individually operational modules from each of the subtasks.

2.2.4 Task 4: ITEC809 Project Report Development [46 hours effort, 6/10/09 – 7/11/09, Weeks 10 – 15]

The purpose of this task is to develop the report outline and final report, which are deliverables for ITEC809. This takes in deliverables from task 1 'System Analysis and Design', as well as learnings from the coding tasks (tasks 2 and 3).

The following sub-tasks which will be used for detailed tracking:

- **4.1. Develop report outline** [12 hours effort] – The deliverable for this subtask is the report outline required for ITEC809.
- **4.2. Write up draft report** [24 hours effort] – This task will produce the first version of the project report suitable for review, and should heavily utilise the artefacts produced during task 1.
- **4.3. Review and revise project report** [10 hours effort] – This task includes reviewing the report produced as part of task 4.2, and making necessary revisions in order to deliver a suitable ITEC809 project report.

2.2.5 Task 5: ITEC809 Workshop Preparation [10 hours effort, 3/11/09 – 9/11/09, Weeks 14 – 15]

The purpose of this task is to develop presentation material and prepare for the final project presentation, which are deliverables for ITEC809.

3 References

Sommerville, I [2007] *Software Engineering 8th Edition*, Addison-Wesley, Harlow, England. Ch. 4 Software processes

Appendix A – Gantt Chart covering ITEC809

ID	Task Name	Work	Start	Finish
28	**Phase 2 - Software Development**	**374 hrs**	**Tue 4/08/09**	**Thu 14/01/10**
29	**ITEC809 Project Development**	**180 hrs**	**Sun 16/08/09**	**Thu 29/10/09**
30	**1. System Analysis and Design**	**56 hrs**	**Sun 16/08/09**	**Thu 3/09/09**
31	1.1. Design server-side data model	8 hrs	Sun 16/08/09	Tue 18/08/09
32	1.2. Design server-side class model	8 hrs	Thu 20/08/09	Sun 23/08/09
33	1.3. Server-side sequence diagram documentation	8 hrs	Sun 23/08/09	Thu 27/08/09
34	1.4. Design client screens	8 hrs	Thu 27/08/09	Fri 28/08/09
35	1.5. Design client-side data model	8 hrs	Fri 28/08/09	Sat 29/08/09
36	1.6. Design client-side class model	8 hrs	Sat 29/08/09	Sun 30/08/09
37	1.7. Client-side sequence diagram documentation	8 hrs	Mon 31/08/09	Thu 3/09/09
38	**System Analysis and Design Complete**	0 hrs	Thu 3/09/09	Thu 3/09/09
39	**2. Coding Iteration 1 - Server Core**	**60 hrs**	**Thu 3/09/09**	**Sat 26/09/09**
40	2.1. Develop data objects	8 hrs	Thu 3/09/09	Fri 4/09/09
41	2.2. Develop database access layer	12 hrs	Fri 4/09/09	Sun 6/09/09
42	2.3. Develop REST service	8 hrs	Mon 7/09/09	Sat 12/09/09
43	2.4. Develop rules and rule engine integration	16 hrs	Sat 12/09/09	Sat 19/09/09
44	2.5. Build code for application server	16 hrs	Sat 19/09/09	Sat 26/09/09
45	**Coding Iteration 1 Complete**	0 hrs	Sat 26/09/09	Sat 26/09/09
46	**3. Coding Iteration 2 - Mobile Core**	**64 hrs**	**Sat 26/09/09**	**Thu 29/10/09**
47	3.1. Develop data objects	8 hrs	Sat 26/09/09	Tue 29/09/09
48	3.2. Develop data access layer	16 hrs	Tue 29/09/09	Mon 5/10/09
49	3.3. Develop integration to REST service	16 hrs	Sun 11/10/09	Sun 18/10/09
50	3.4. Develop status query screen	12 hrs	Sun 18/10/09	Sat 24/10/09
51	3.5. Develop status entry screen	12 hrs	Sat 24/10/09	Thu 29/10/09
52	**Coding Iteration 2 Complete**	0 hrs	Thu 29/10/09	Thu 29/10/09
53	**4. ITEC809 Project Report Development**	**46 hrs**	**Tue 6/10/09**	**Mon 2/11/09**
54	4.1. Develop report outline	12 hrs	Tue 6/10/09	Wed 7/10/09
55	**ITEC809 Report Outline Due**	0 hrs	Fri 16/10/09	Fri 16/10/09
56	4.2. Write up draft report	24 hrs	Wed 7/10/09	Sat 10/10/09
57	4.3. Review and revise project report	10 hrs	Sat 31/10/09	Mon 2/11/09
58	**ITEC809 Project Report Due**	0 hrs	Mon 2/11/09	Mon 2/11/09
59	**5. ITEC809 Workshop Preparation**	**10 hrs**	**Tue 3/11/09**	**Mon 9/11/09**
60	Prepare for Workshop Presentation	10 hrs	Tue 3/11/09	Mon 9/11/09
61	**ITEC809 Workshop Presentation**	0 hrs	Mon 9/11/09	Mon 9/11/09