Introduction

Purpose of document

This document aims to provide an architectural overview of the Crowd/tidbits system, and to describe the different aspects of the system with regard to architectural decisions made on the structure of the system.

Scope of project

The system needs to provide a mobile application allowing users to upload messages that are relevant to a specific location. To this end, an iOS/Android application, a back-end HTTP server, and a persistence unit.

This section discusses the software architecture requirements | that is the requirements around

the software infrastructure within which the application functionality is to be developed. The

purpose of this infrastructure is to address the non-functional requirements. In particular, the

architecture requirements specify

• the architectural responsibilities which need to be addressed,

• the access and integration requirements for the system,

• the quality requirements, and

• the architecture constraints speci\_ed by the client.

1 Access and integration requirements

This section discusses

1. the requirements for the di\_erent channels through which the system can be accessed by

people and systems, and

2. the integration channels which must be supported by this system.

1.1 Access channels

This section speci\_es the di\_erent channels through which users will be able to access the system

services.

1.1.1 Human access channels

Specify how humans will access the system. e.g. whether thick application clients or thin web

clients need to be provided, any standards compliance and any dvices or software which needs

to be supported.

1.1.2 System access channels

If required, specify how other systems will access the system.

1.2 Integration channels

Specify the other systems this system needs to integrate with, and requirements around these

integration channels.

2 Architectural responsibilities

Specify any additional architectural responsibilities which the system will have to address like

persisting domain data, storing documents, providing an event infrastructure, providing an exe-

cution environment for processes, . . .

2

3 Quality requirements

The quality requirements are the requirements around the quality attributes of the systems and the

services it provides. This includes requirements like performance, scalability, security, auditabilty,

usability, and testability requirements.

3.1 Scalability

Specify quanti\_ed scalability requirements.

3.2 Performance requirements

Specify quanti\_ed performance requirements.

3.3 Maintainabilty

Specify the maintainability requirements including where change is expected and bounds to the

e\_ort required to apply changes to the system.

3.4 Reliability and Availability

Specify reliability and availability requirements.

3.5 Security

Specify security requirements

3.6 Monitorability and Auditability

Discuss monitorability and auditability requirements including what aspects and through which

channels the the system should be monitorable, which information should be captured for au-

diting purposes and how the audit logs are made accessible.

3.7 Testability

Specify testability requirements.

3.8 Usability

Specify usability requirements.

3

3.9 Integrability

Specify integration requirements including required integration channels, protocols and quality

requirements on integration channels

4 Architecture constraints

Specify any architectural constraints the client has speci\_ed, e.g. to use a particular relational

database, programming language or framework