Project overview and SRS

Term 3 project: Business Intelligence system

AUTHOR: PATRIK BÄCKSTRÖM

VERSION: 1.0

Table of contents

Introduction

Overall Description

System requirements specification (SRS)

Products

References

Points Of Contact

1. Introduction

This document contains an overview of requirements and functions specifying and clarifying the intention and usage of *'Adapa stock watcher'*.

This document consists of goals and objectives, scope, product overview, requirements, quality attributes, and contact information, and describes how you access the software.

1.1.Purpose

The purpose of this document is to explain all features and functions available through this software. It shall also map functionality to what requirements they may fulfil. This document is intended to the supervisor and examiner assigned to Group Adapa, as well as developers and end users of ASW.

1.1.1.Goals and objectives

The main objective of this project is to implement a Business Intelligence system (Adapa stock watcher). Specifically, that includes to develop software which collects relevant stock data from four sources, stores fetched data in a data warehouse and enables that data through three interfaces developed towards different platforms.

Adapa have chosen to aim ASW toward three markets; Nasdaq, London Stock Exchange, and DAX (Deutscher Aktienindex). It will also collect ticker symbols used to request stock specific data and excludes those symbols not available through used sources. That is; requests where returned data does not contain what Adapa consider 'all necessary data' (found in *Software Design Document, section 6.3 JSON structure)*.

Requirements to fulfil in order to achieve mentioned objectives and goals is listed in section *3 Requirements*.

1.1.2.Source mapping

The following section shows what sources ASW fetches data from, what market it relates to, and what technology is used to perform the requests.

ID	Market or data source	Source	Technology	Data type
1	London Stock Exchange	Google Finance	Google Action Script (GAS)	**Stock**
2	DAX	Yahoo Fincances	Yahoo Finance API	**Stock**
3	Nasdaq	MarkitonDemand	Direct Erlang httpc request	**Stock**
4	*List of ticker symbols*	*Nasdaq's Official webpage*	*Direct Erlang httpc request*	***Symbols***
5	*List of ticker symbol*	*myinvestorhub.com*	*Direct Erlang httpc request*	***Symbols***

1.2.Scope

ASW will be regarded complete when required documentation, planning, implementation, testing, low-level design, and high-level design is satisfactory. Also required to be considered complete, is that all interfaces meet requirements specified in *3 Requirements*, and all non-mandatory functions mentioned in section *2.2 Product functionality* is implemented. Requirements are collected from Group Adapa's own demands, as well as requirements set out in the Software Architecture for Distributed Systems specifications.

All parts shall be carried out by members of Group Adapa, both with support from courses given in the SEM curriculum, and individual knowledge and experience achieved in previous projects.

1.3.Definitions, acronyms and abbreviations

ASW - Adapa stock watcher
SEM - Software Engineering and Management programme
ETL - Extract, Transform, Load
GIT - Version control service. Achieved through BitBucket.org
OTP - An Erlang standard set by Ericson, allowing module supervision and error handling.

2. Overall Description

The following section provides a high level description of the entire application. It describes the product perspective, functionality, characteristics of an expected user, constraints, assumptions, and dependencies.

2.1.Product perspective

ASW is available through three different interfaces, allowing access from Android mobile devices, Windows computers, and many other devices using modern web browsers. They were developed and are maintained to resemble the others in functionality, quality, design, and ease of use. Interface design has also been inspired by similar webpages and applications to help new users adapt to ASW. Interface research can be found in *UI Research Report*.

ASW collects data from free sources with a fully automated data fetcher that stores data externally or locally, through CouchDB's RESTful interface. With CouchDB installed, our ETL, when run, will set up CouchDB with all necessary databases and views to support it whilst running. Ticker symbols related to available markets are also fetched when CouchDB is set up, and thereafter dropped and updated once every day to catch new symbols. Stock data is collected once every hour as long as the ETL is running and the current stock market is open.

Setting up the software shall only be possible by administrators who possesses access to online repositories, and therefore, all documentation regarding setup is not of relevance to end users.

2.2.Product functionality

- ASW can be setup with as many stock markets as existing sources can provide.
- Webpage provides cross-platform functionality.
- Mobile application support backwards compatibility down to Android version 10.
- Interfaces provide smooth and quick search and filter functions.
- Interfaces provide interactive graphs.
- Users can add individual stocks to their watch list.

List of functions available to users

Function	Mobile	Desktop	Webpage
List view	X	X	X
Detailed stock view	X	X	X
Market index view	X	X	X
My stocks	X	X	X
RSS news feed	X	X	X
Multiple users (NOT MANDATORY)	X		
Stock related news (NOT MANDATORY)	X		

2.3.User characteristics

2.3.1.Characteristics of Administrators

Administers have to understand how to:

- clone repositories from a GIT service,
- install CouchDB,
- install Erlang and start OTP applications, OR
- alternatively compile the ETL with supplied makefile (UNIX only), and
- modify Erlang definitions files to direct requests to a server.

2.3.2.Characteristics of End users

End users have to be familiar with:

- navigating to a webpage, OR
- installing software from Google Play, OR
- installing software on Windows systems, and
- basic knowledge of how stock market analysis works.

2.4.Constraints, assumptions, and dependencies

- Internet connection.
- JavaScript enabled web browser.
- Mobile users have got an Android phone able to interpret API level 11, or greater.

2.5.Possible improvements

As always, there are a lot of possible improvements floating. Some improvements was being considered requirements at an early stage. For example, allowing multiple users was not considered until very late into the project, and is therefore not yet developed for the web page.

Possible improvements Adapa have discussed are:

- Allow multiple users in all interfaces.
- Include more markets.
- Supply more RSS feeds and allow better selection and sorting.
- Show most common currencies together with market indexes.
- List todays highs and lows.
- Supply expert market estimations and recommendations.

3. System requirements specification (SRS)

This section contains requirement lists only. To see current progress, read *Quality Management report*.

ASW is a software highly dependant on automated services running in the background, not interfering with user. End user's possibility to interact with the system is kept to what is necessary to complete requirements. Therefore there might seem to be very few requirements aimed towards the user. However, many more requirements reflects quality of the ETL and database, which is important to guarantee uptime, reliability, and performance.

3.1.Functional requirements

- User shall be able to analyse market index details by chart and numbers.
- User shall be able to analyse stock details by chart and numbers.
- User shall be able to view stock in three different views.
- User shall be able to search and filter stocks from selected market.
- System displays news about each available stock.
- System shall supply user with RSS news feeds for available markets.
- User shall be able to follow desired stocks in all applications.

3.2.Quality attributes

- Data collected from sources shall be accessible in applications maximum five minutes after being fetched.
- Applications shall only receive stock data from backend.
- One view shall be candlestick chart.
- ETL shall be implemented in Erlang.
- User shall be able to access data via three platforms (mobile, desktop, web).
- System shall use at least four different data sources.
- Database shall only consist of well formatted data.
- Database shall contain data up to five years old.
- Database shall not use any SQL language.
- ETL shall discard data not up to standard.
- ETL shall be packaged and run as an OTP application.
- ETL shall be automated, meaning that once started, no more interaction is necessary.
- ETL shall set up database properties necessary to run.

4. Products

4.1.How to run the products

4.1.1.Web page

1. Clone BitBucket repository.
2. Go to [GIT_FOLDER]/webdev/ and launch index.html.

OR, go to http://83.254.83.56/src/webdev/index.html.

4.1.2.Mobile application

1. Go to Google Play.
2. Search for Adapa stock watcher.
3. Download and run application.

4.1.3.Desktop application

1. Clone BitBucket repository.
2. Go to [GIT_FOLDER]/desktop/ and run setup.exe.

5. References

CouchDB, http://couchdb.apache.org.
Erlang, http://www.erlang.org.
Bitbucket, https://bitbucket.org.
Android API level 11 support, http://developer.android.com/reference/packages.html.

6. Points Of Contact

Name	Role	Email
Patrik Bäckström	Project Manager	gusbacpa@student.gu.se
John Burchell	Software Designer	gusburchjo@student.gu.se
Nico Boh	Technical Expert	gusnicobo@student.gu.se
Carl Berglund	UI Designer	gusbercaal@student.gu.se
Ludvig Gee	Architect	gusgeelu@student.gu.se
Michael Thompson	Quality Manager	gusthompmi@student.gu.se
David Gregory	Supervisor	david_gregory60@hotmail.com