**Chapter 1**

**Introduction**

**1.1 Overview**

The purpose of this MSc dissertation is to implement an interactive, user-friendly visualization technique that represents the update history of Database Wiki project structure and values. Database wiki is an open source project that combines wiki approach with advantages of relational database systems. This project focuses on visualization that works on Database Wiki however it could be adapted to any other similar project.

**1.2 Motivation**

The Wiki approach has gained huge popularity in many different fields during the last decade. According to [1] - Wiki is the set of web sites that are connected with hyperlinks. The main feature of the wiki web is that information can be updated, added or even deleted by many users not only by one or a few authors. The main issue with this approach is that often anyone or a large group of users have rights to change data. Inevitably some updates will be incorrect, subjective or even vandalistic, such as mass deletion of the content, insertion of information which is offensive, uncensored or off-topic [2]. In order to deal with updates like these Wiki pages need an efficient way of restoring information to the previous version which is correct. That’s why all the versions of information, not only the current one, should be saved in the database. Some solutions have been proposed that store all the versions of the information in a hierarchical structure where the changes are saved as nodes of the tree [3] [4] (this technique will be referred to as History Tree Archiver further in this paper). This way the information is consistent and compact. These techniques can also be beneficially applied to provenance management. A good example of this would be scientific databases when the origin of certain data can be very important.

The Wiki approach has contributed enormously to the Web as we know it today. Some of its projects have grown enormously in recent years, for example Wikipedia, which is the largest encyclopedia in the world today. However with the scale of wiki project growing in such enormous speeds new challenges arise. How to track all the changes of the content and decide which of them should be undone? This looks like an easy tasks when there are only a few pages which are updated only by a few users, but what to do if the number of updates, authors and pages are huge, for example in a project like Wikipedia? How to find acts of vandalism efficiently, moreover how to track and identify users that are constantly making inaccurate or subjective changes that should not be interpreted as vandalism acts when done only once but are harmful when performed systematically. Manual checking of information would demand huge amounts of labor so more efficient methods are needed. One of the efficient approaches to analyze data of changes which is offered by [2] is to visualize an updates history flow. This way a large amount of data can be presented to a user in a form which is much more practical in terms of analyzing and interpreting it. Certain patterns in graphical representation of data help to identify acts of vandalism or misleading information in updates much more easily so the permissions for certain users could be changed to prevent such updates in the future.

This project uses two types of graphs to visualize the changes made to data objects in the database. The first type is History Tree Archiver and the second which will be referred as Structure Tree further in this paper (variation of History Tree Archiver that presents only one version of tree at a time). We believe that these approaches will provide new points of view towards database update data and it will increase the understandability of how changes to the data are made in intensively updated wiki systems.

**1.3 Purpose**

The main purpose of this project is to implement visualization techniques that represents history changes of the Wiki Database in user friendly, interactive and informative way. This purpose can be divided in two main stages. In first stage appropriate solutions is found and implemented and in the second part they are evaluated and refined if necessary. To achieve desired features of visualization techniques - two different tree type structures are visualized to represent as many features of the update history as possible. The first structure is History tree Archiver displays all of the database versions at once. The main purpose of this visualization technique is to display number and frequency of changes made in database quickly without too many details. These trees can grow large as all the values are represented at once. Another technique that is implemented in this project is Stucture tree visualization. This visualization tool shows the structure of one version at the time and provides a possibility to change displayed version interactively. Main purpose of Structure tree visualization is to get more detailed information about how changes were made, by whom and how the overall structure of the tree changed during some time period. These trees usually are smaller than history trees however they are much more interactive.

After implementation is completed project is evaluated in two main criteria. The first one is the speed of performance. These evaluations are done by different timing tests. The second criterion is user friendliness. Precise evaluation of user friendliness may be hard to achieve however it is made by doing a survey about usability of the visualization system and collecting some feedback from peers. After evaluation stage is done improvements to the project will be made if necessary.

**1.4 Outcome of the project**

Two visualization techniques were implemented in this project.