

MVC ARCHITECTURE DRIVEN DESIGN AND AGILE IMPLEMENTATION OF A WEB-BASED SOFTWARE SYSTEM

Sandhya Prakash
Vancouver, B.C., Canada

Ashok Kumar
University of Louisiana at Lafayette, USA

Ravi Bhushan Mishra
Indian Institute of Technology (BHU), Varanasi, India

ABSTRACT

This paper reports design and implementation of a web based software system for storing and managing information related to time management and productivity of employees working on a project. The system has been designed and implemented with best principles from model view controller and agile development. Such system has practical use for any organization in terms of ease of use, efficiency, and cost savings. The manuscript describes design of the system as well as its database and user interface. Detailed snapshots of the working system are provided too.

KEYWORDS

Model view control, agile system, time management, web based system, database, user interface, MySQL, struts.

1. INTRODUCTION

Tracking and managing productive time spent on projects by software developers and other employees is a very important task for any organization. This work is focussed on design and development of a timesheet management system (TMS). The proposed system is a web-based, multi-user application that is designed to be used beyond geographical boundaries and intended to help employees to manage their work hours by keeping track of their daily and weekly productive hours (i.e., time spent on projects). The information is stored in a central database system. Computerized information collection, storage, and retrieval brings accuracy to the time management system.

TMS is a model view control based system. It generates reports for employees and helps to create, update and delete projects and calculates salaries for the employees by tracking their daily work hours. TMS was mainly developed to accurately track employees' work hours, to maintain computerized attendance system, and based on this information, to calculate salaries for the employees. The software was tested thoroughly to understand project costs, client bills, payroll, and work productivity. TMS helps increase productivity among employees and it increases accountability across an organization.

The remainder of the paper discusses project requirements, selection of design and implementation methodologies, details of implementation, and snapshots of actual implementation.

2. DESIGN AND IMPLEMENTATION

A successful design of the presented software system is based on well described user and system requirement specification, selection of appropriate design and implementation methodologies, actual implementation, testing, and deployment. The development begins with understanding the process by presenting a requirement document that specifies the purpose of the proposed TMS system and the functions that it must perform in order to meet the requirements. TMS system needed a lightweight framework of development to accommodate possible changes in functional and technical specifications, to accommodate easy interactions among developers, and to produce a rapid prototype of the system. Cost saving and flexibility were the key criteria in our selection of agile methodology. The project was broken down into smaller pieces of functionality and frequent iterations in all aspects of its development such as planning, analysis, design, development, testing, and integration were entertained. Also, all stakeholders of the system such as executives, managers, customers, software developers and testers were involved in the process and the system was allowed to be flexible enough to accommodate requests for any reasonable changes from any stakeholders.

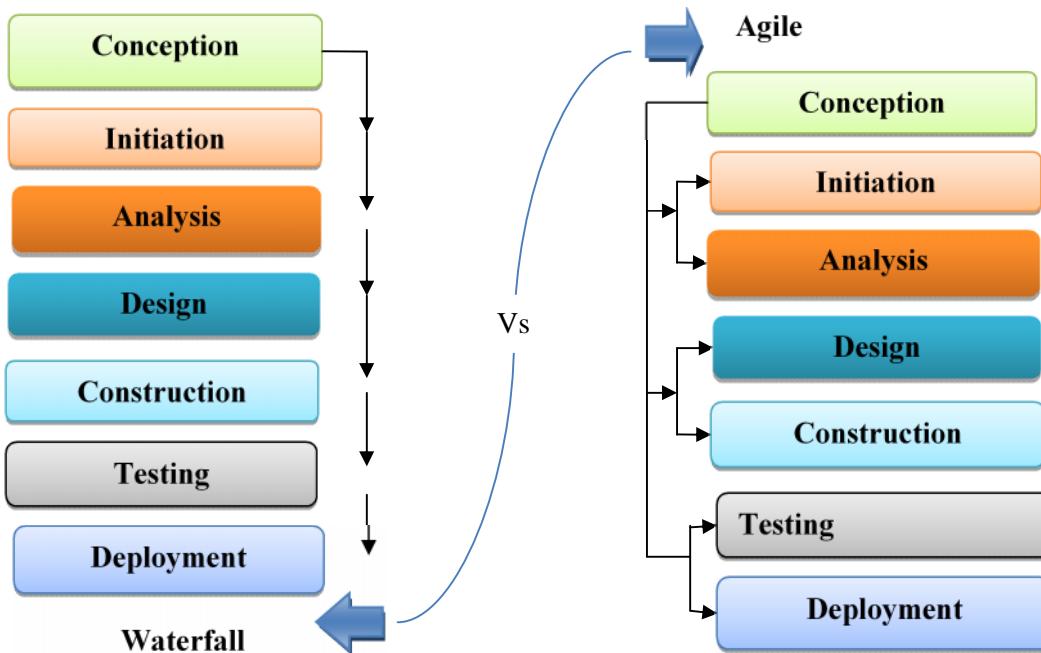


Figure 1. Agile versus Waterfall models

2.1 Agile Model

The TMS system used agile model of implementation which is described next. Agile model of software development differs from the traditional waterfall model as shown in Figure 1.

An agile process produces greater customer satisfaction by quickly producing an initial prototype of system and allows customers to have a feel for the system and most importantly it allows customers to start giving them inputs early in the development rather than in the late stages of development. Requirements are allowed to change in an agile process and yet an agile process, if implemented properly, is completed in a given time frame. An agile process typically needs to have team members with high accountability and excellence in providing timely inputs for improving the system, and incorporating changes in the system as needed. Team members should include not only developers and other technical executives but also representatives of end users. Such team is typically empowered to make decisions which are crucial for the success of an agile process.

An agile process is typically light weight and changes to the system-under-development are considered a rule rather than an exception. Requirements can evolve over a time period and an agile process accommodates such changes. This is one of the most significant advantages of an agile process over traditional software development processes. Agile process is iterative, it allows projects to be developed incrementally, and it greatly reduces the risk of the end result to be significantly different from what was envisioned at the beginning of the project. Project deliverables and meeting customers' expectations are given much higher weight than rigorous documentation.

It is clear that an agile process needs more frequent interactions among stakeholders and team members and just-in-time approaches are typically used for such purposes. Frequent releases and deliveries of software system is done in the agile methodology in order to seek customers' feedback and in order to allow software development (code) changes. In an agile process, a software development team not only implements what the end user asks for but also provides the end user with realistic view of which features can be included in the product a given timeline and which cannot. Such communications, feedbacks, and adjustments help the agile process to achieve high degree of success in comparison to traditional waterfall based approach of software development.

2.2MVC Architecture

The TMS system used the model-view-controller (MVC) design pattern and Struts framework. Using the MVC approach, an application is divided into three functional parts named as model, view and controller. The model part represents the business logic of the TMS software applications as well as access to relational database. The view is the visual interface provides a medium to web users to interact with the system. View source codes for the TMS system are created using java server pages (JSP) technology. The controller consists of one or more servlets, configuration files and Struts actions files. MVC pattern is shown in Figure 2.

Generally, in model2 web applications the web server (e.g. Tomcat in the case of TMS) contains view and controller both at application run time while data storage is done in third tier named as Model. The view part represents user interface and it is created using JSP files that generally do not contain any business logic in order to ease the rebuild process and to reduce the complexity of applications internal dependency.

The controller part of MVC architecture is a component on web server that process all incoming requests from user via user interface to a single servlet instance. Due to heavy load of processing requests this controller is also known as fat controller.

Java beans are known as model component in an MVC architecture. To process the requested business logic, the action servlet selects and invokes one or more actions to process the request at a time. Action generally creates or modifies java beans which is stored as a request or session object and changes the state of a web application when a user interacts with the UI and requests something. The RequestDispatcher.forward() method of servlet API is called by the action to generate the next requested page by the user instead directly producing the requested page.

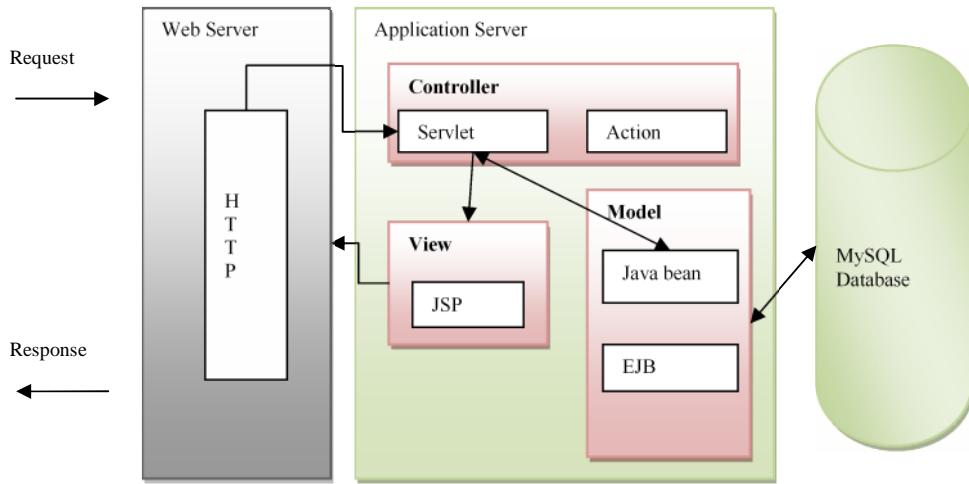


Figure 2.MVC Design pattern overview

2.2.1 Advantages of MVC Design

The implementation of model-view-controller in three different dynamic parts has many direct benefits. For example it is easy to delegate the development work within developers or team members and distribute the total effort and MVC model ensures that changes in one web application will not affect the other web application. As an example web designer and web developer can work independently or a coder working on business logic can work independently of the professional working application flow control.

By implementing the MVC architecture it becomes easy to prototype the work by following very simple steps like to create a prototype of a web application that access several work stations-based programs. By making little changes to the configuration files or renaming the server contents only, one can implement production level programs that can run on different platforms without having to rewrite the source code.

Migration of legacy programs has become easy since the model and controller are totally separated in MVC and it makes tailoring the user category or platform much simpler. MVC has also great contribution in simplifying the scalability problem of software programs of large size applications and making modification and maintenance of applications easier through clear separation of tasks.

2.3 USE CASE DIAGRAMS FOR THE TMS SYSTEM

After deciding most suitable development methodologies for TMS software design, the process begins with a simple use case diagrams that represent interaction between the system and its users without being trapped into details. Use case diagrams for TMS are presented in Figure 3

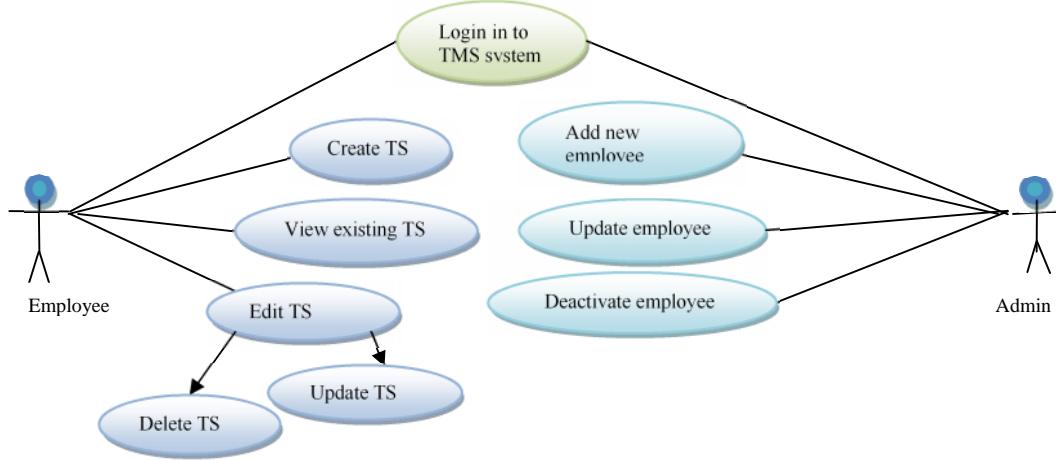


Figure 3. Use-Case diagram of interaction between employee, admin and system

2.4. REPRESENTATION OF BUSINESS FUNCTIONS FOR THE TMS SYSTEM

Data flow diagrams are used by system analysts and software developers to represent the functional requirements of the system and to model the process. Data flow diagram is considered as one of the efficient modelling techniques for eliciting and representing the processing requirements of a system. Data flow diagram for the TMS system is shown in Figure 4 and it captures business functions for the proposed system. It follows the rules of modern flowcharts.

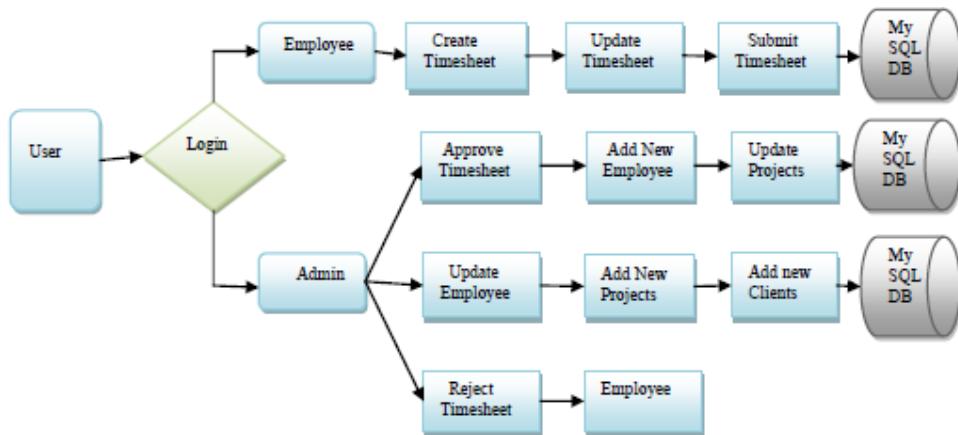


Figure 4. Data flow diagram of the TMS system

2.5. CLASS DIAGRAM FOR THE TMS SYSTEM

The class diagram for the TMS system is shown in Figure 5. It consists of attributes(data) and operations (behaviors) such that attributes are implemented as fields and class operations are implemented as method. The entity-relationship for TMS is shown in Figure 5 and Figure 6.

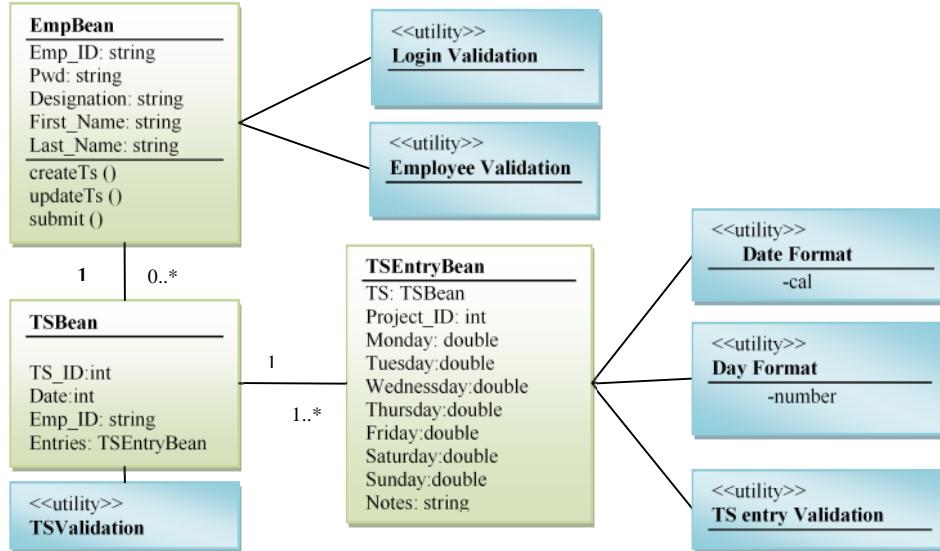


Figure 5. Class diagram with attributes and operations (incorporating inheritance)

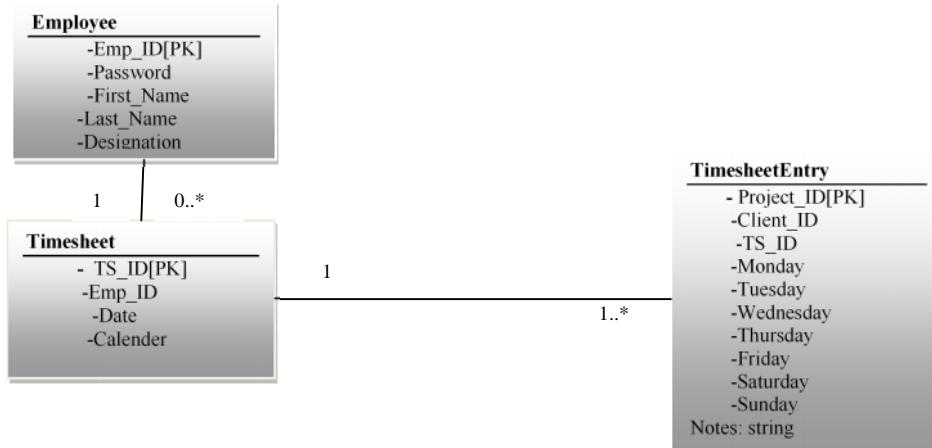


Figure 6. Entity relationship model for TMS

2.6. DATABASE AND ORM FRAMEWORK (HIBERNATE) IN TMS

Database system used in TMS is a relational database management system (RDBMS) and it is a collection of tables to store data. The database design process starts with information gathering and understanding the functional requirements in the project. The document should give a concise summary of all users' requirements, not just a collection of individuals' requirements, as the intention is to develop a single shared database.

The requirements should not describe how the data is to be processed, but rather what the data items are, what attributes they have, what constraints apply and the relationships that hold between the data items. Hibernate has been used in TMS as it provides ORM functions very well as an abstraction layer between Java and MySQL database. The intent here is that Java will be able to work internally with the objects defined as they desire, while the database is designed in a purely relational.

In TMS system, key users of the system are employees and the system was built to track the productive work hours of the employees. The main tables in database are shown in Table 1, Table 2, and Table 3. The attributes of the table are self-explanatory.

Table1- Employee Details

ColumnName	DataType	VariableLength	Null Value
Emp_Id	int	10	Not null
Last_Name	varchar	20	Not null
First_Name	varchar	20	Yes
Emp_Designation	varchar	30	Yes
Email	varchar	20	Yes
Cell_Phone	varchar	20	Yes
Home_Phone	varchar	20	Yes
Photo_Id	image	30	Yes
Street	varchar	50	Yes
City	tinyint	1	Yes
Zip	varchar	20	Yes
State	tinyint	1	Yes
Nationality	varchar	30	Yes
Visa_Status	varchar	50	Yes
Sex	varchar	20	Yes
Birth_Date	date	50	Yes
Emp_Type	varchar	30	Yes
Dept_Id	int	20	Yes
Emp_Memo	varchar	300	Yes

Table2- Login

ColumnName	DataType	Length	NullValue
Emp_Id	varchar	30	Not null
Password	varchar	30	Not null
Emp_Type	tinyint	1	Not null

Table3 - Projects Details

ColumnName	DataType	Length	NullValue

Proj_Id	Int		
Proj_Name	varchar		
Client_Name	varchar		
Proj_Start_Date	varchar		
Proj_End_Date	varchar		

Hibernate was used in the TMS system to map object oriented domain model to relational database. Hibernate framework simplifies the development of java application to interact with the database. Hibernate is an open source, light weight, object relational mapping tool. An ORM simplifies the data creation, data manipulation and data access. It is a programming technique to map the objects to data stored in a database in the form of tables.

There are many advantages of using hibernate framework for accessing data from a database in a java web application. Some key advantages of why Hibernate was used in TMS are listed here. First, Hibernate is an Open source software and very light weight. Second its performance is very high because cache memory is utilized internally. Usually Hibernate framework uses two types of cache memory first level cache and second level cache. First level cache is enabled by default. Third benefit is Hibernate uses a database-independent query language known as Hibernate Query Language (HQL) which is an object oriented version of SQL. It generates the database independent queries. It means there is no need to change the query when changing the database. As a result, database specific queries are not required. Hibernate supports automated table creation which means database tables can be created automatically so no there is need to create tables manually. Hibernate offers a technique to fetch data from multiple tables without complexity. Hibernate is also well known for its excellent stability and quality, it allows persistence, and it scales well.

2.7. STRUTS IN TMS

TMS uses an MVC architecture driven design and its implementation needed a web application framework that could use Java Servlets and facilitate adherence to the MVC architecture. For this reason, Struts were found to be a perfect match for the TMS system. For an example, the strut based design of the login process in MVC is shown in Figure 9 and its processing details are shown in Figure 7.

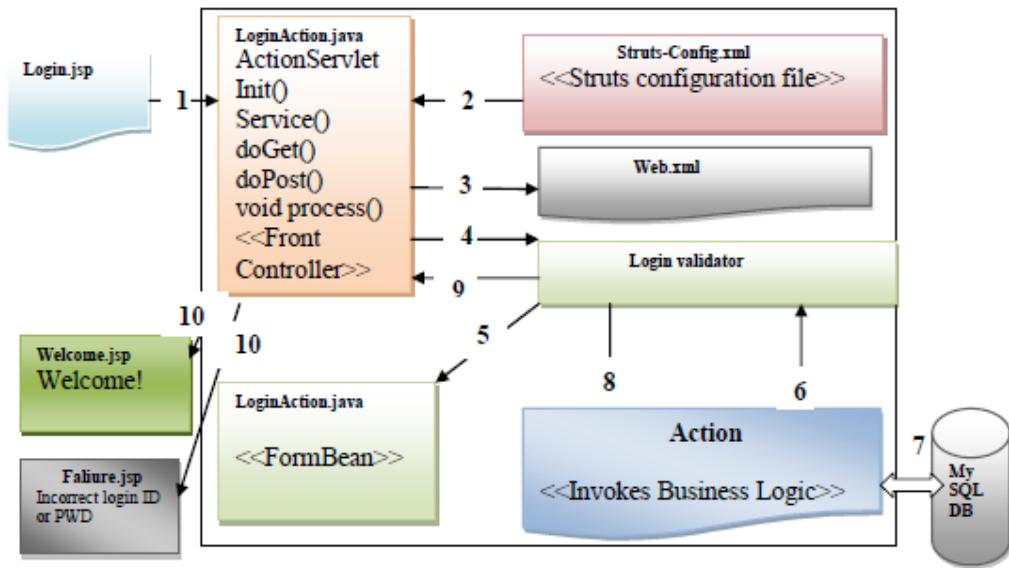


Figure 7. Strut based design of login process architecture in TMS

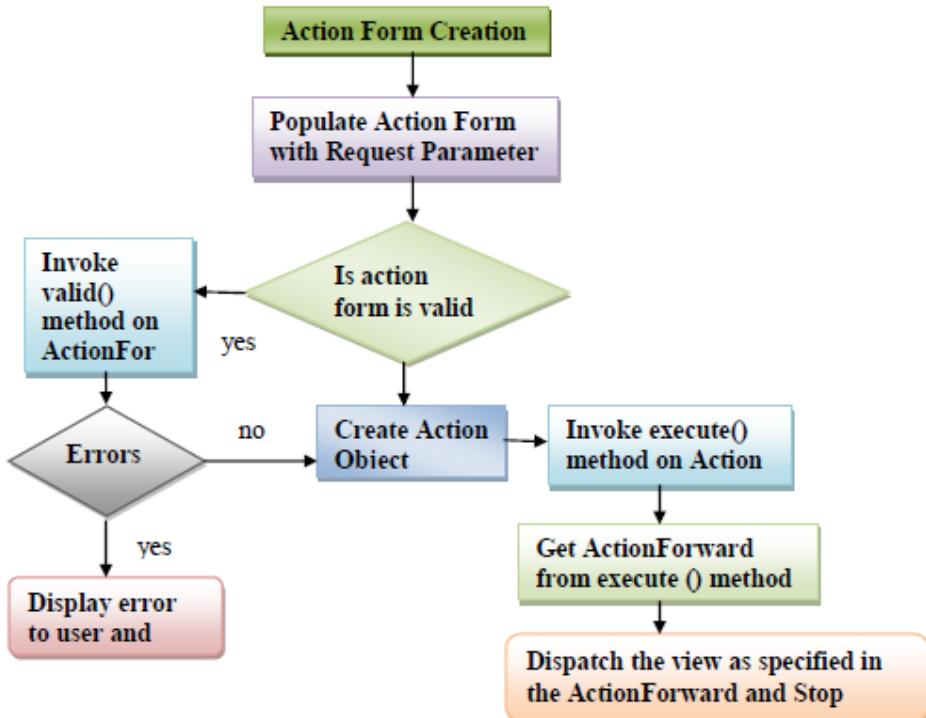


Figure 8. Processing details

2.8. SNAPSHOTS OF WORKING TMS SYSTEM

The TMS system used extensive testing including unit-tests to ensure proper functionality and robustness of code. Classes that were difficult to test were refactored and deep hierarchies were avoided in the design. The system underwent thorough testing and it was found to be fully functional. Some of the snapshots of the system are shown in Figure 9 through Figure 20.

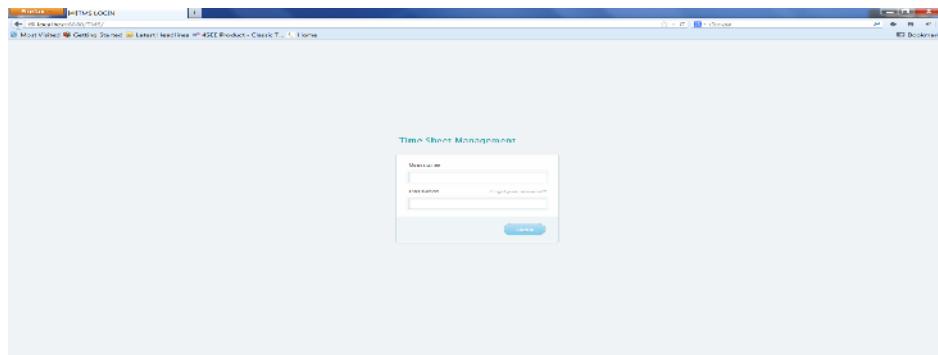


Figure 9. Home page of time sheet management system

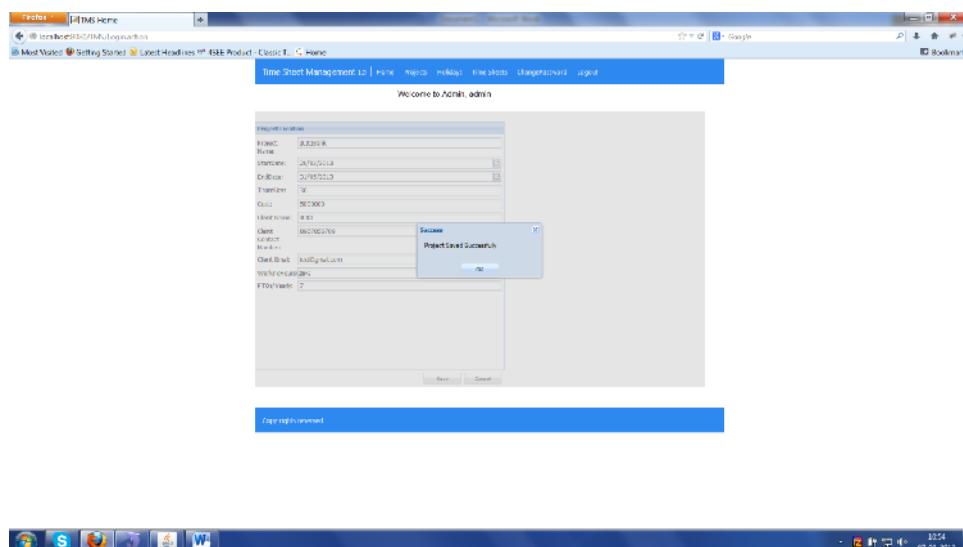


Figure 10. Storing information of a new project

Admin Home page:

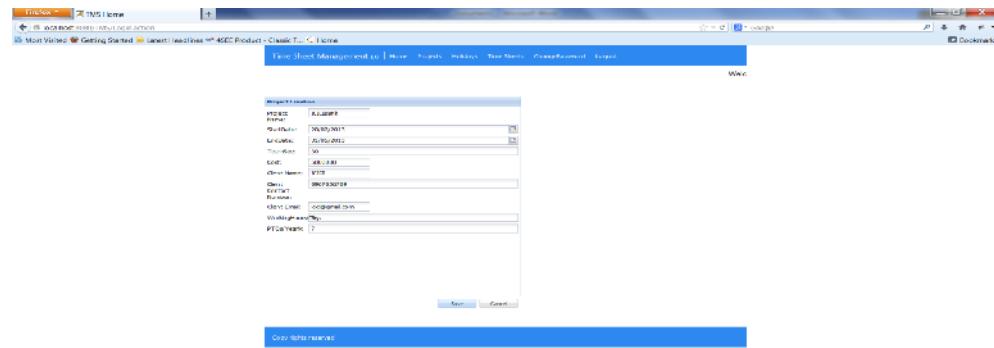


Figure 11. Adding new projects

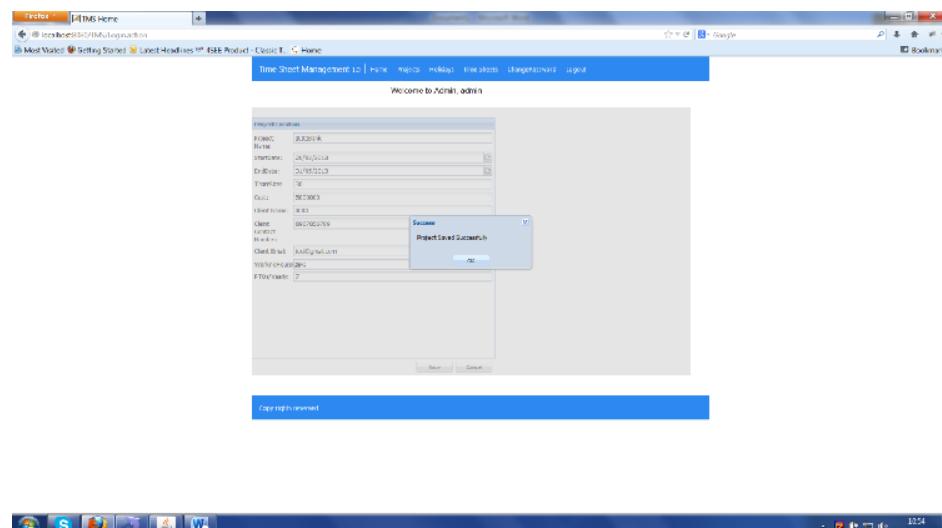


Figure 12. Storing information of a newly added project

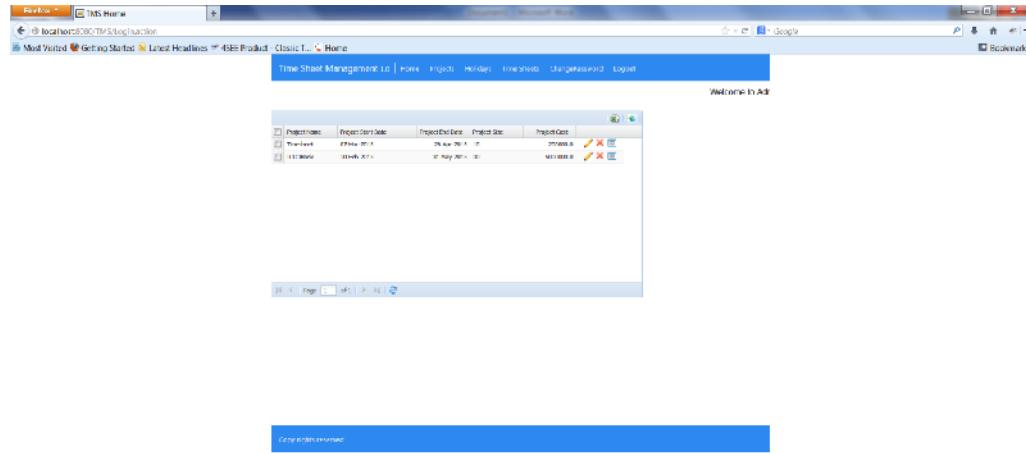


Figure 13. Updated display of a newly added project

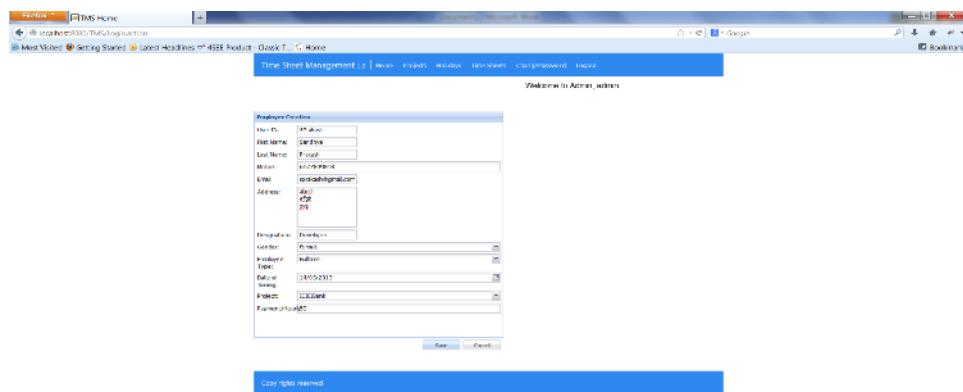


Figure 14. Addition of a new employee to the system

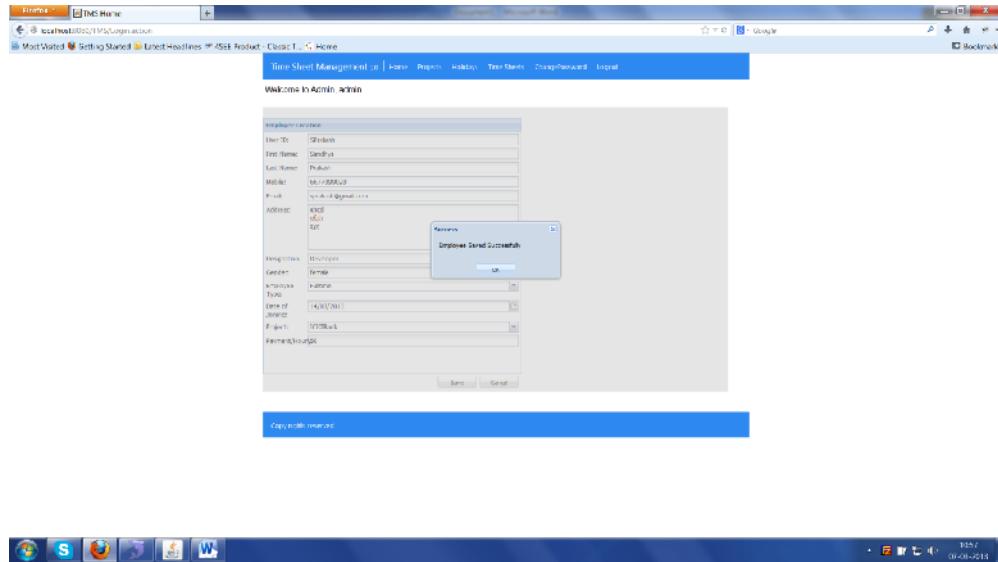


Figure 15. Storing information related to an employee

Again Login as Administrator to ‘Approve/Reject’ Employee Timesheet:

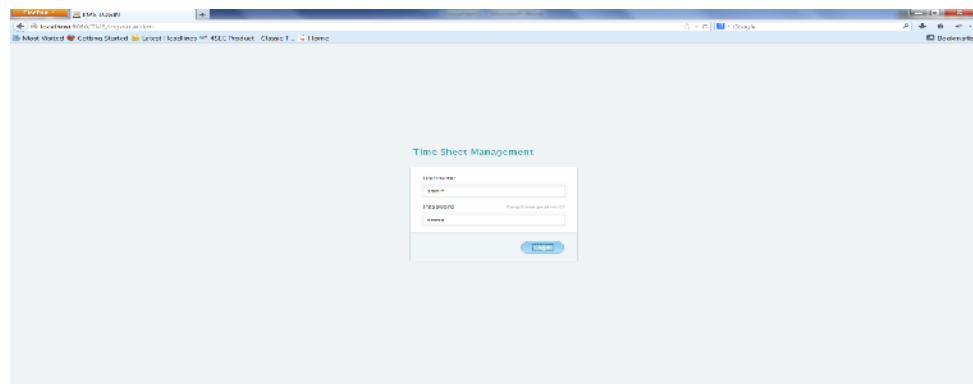


Figure 16. Administration login for approval or rejection

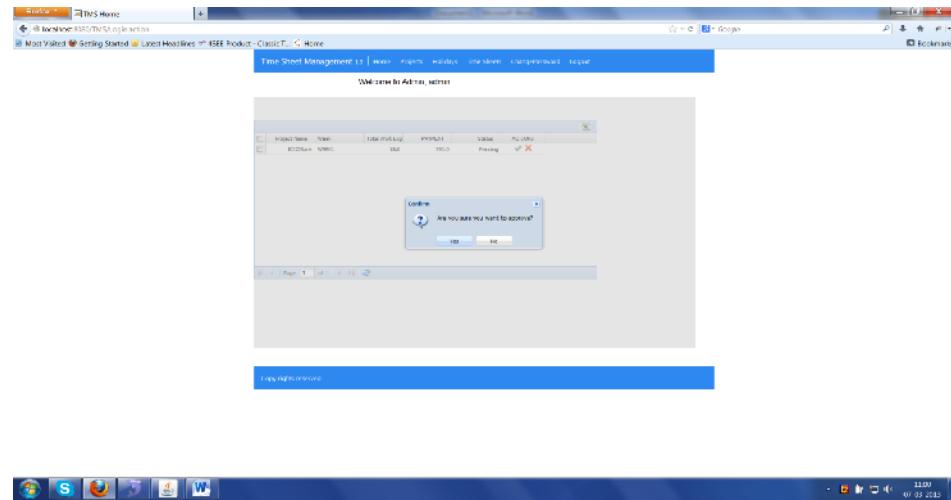


Figure 17. Administration approval or rejection of timesheet

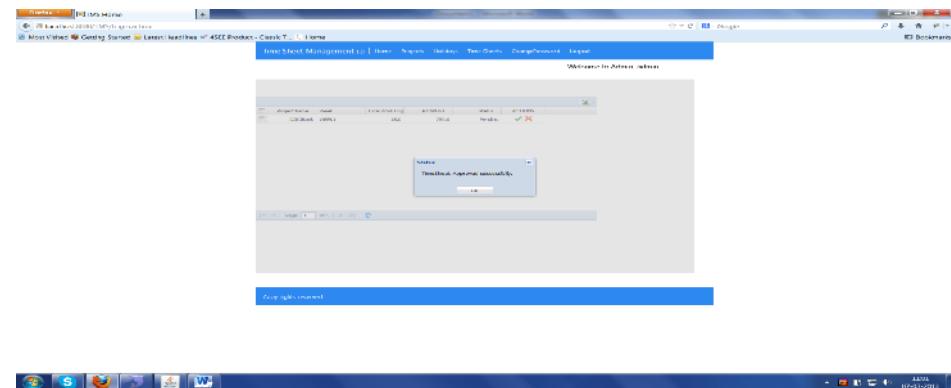


Figure 18. Display of updated time sheet

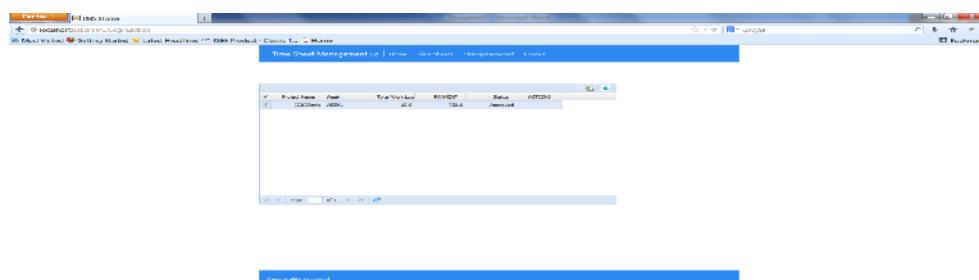


Figure 19. Employee's view of an approved or rejected timesheet

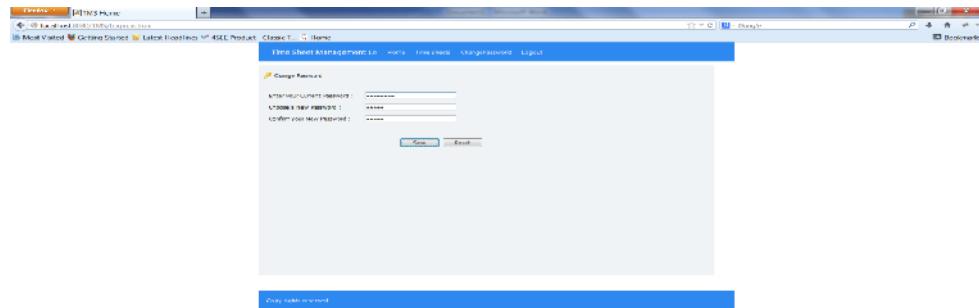


Figure 20. Password change option

2.9. RELATED WORK

The work in [1] is the seminal work on agile development and it has been widely cited and used. The book in [2] provides an in-depth coverage of MVC architecture. The Struts framework is described in [3] in detail. A web-based time management system was reported in [4] using a client-server model. The works in [5] and [6] are somewhat, but very remotely, related as they use struts and hibernate too.

3. CONCLUSIONS

This paper presented design and implementation of a software system for time management by employing the best practices from model view controller based software architecture as well as agile methodology based implementation. The proposed idea has been successfully implemented and tested thoroughly, and it has practical use in both industry in academia.

ACKNOWLEDGEMENTS

The first author would like to acknowledge her teammates.

REFERENCES

- [1] Kent Beck, et al., "Manifesto for Agile Software Development", 2001, available at <http://agilemanifesto.org>.
- [2] Frank Buschman, et. al., "Pattern-Oriented Software Architecture Volume 1: A System of Patterns", 2000, John Wiley & Sons.
- [3] Struts, Apache Struts Project, <http://struts.apache.org/>
- [4] Z.M. Udin, et. al., "Development of a Web-based Academic Time Management System", Proc. Sixth International Conference on Computer Supported Cooperative Work, 2001, pp. 572-574.
- [5] Jing Yang et. Al., "Management System for Scientific Research Projects based on the Combined Framework of Struts and Hibernate", Proceedings of the 2nd International Conference on Artificial Intelligence, Management Science, and Electronic Commerce, 2011, pp. 5619-5622.
- [6] Li Zhang, et. Al., "Design and Implementation of e-online Tourism System Based on the Hibernate Framework", Proceedings of the International Conference on Electrical and Control Engineering, 2011, pp. 2963-2966.

Authors

Sandhya Prakash received her Bachelors and Masters degrees in Computer Science. Her areas of interest include Internet Computing, Software Engineering, Databases, and User Interfacing.



Dr. Ashok Kumar is an Associate Professor in the School of Computing and Informatics at the University of Louisiana at Lafayette, USA. Dr. Kumar obtained his Ph.D. in 1999 and worked for four years in industry before joining academia full time. He has published extensively in refereed journals, conferences, and book chapters. He has served on the program committees of several conferences. His areas of interest span hardware design for low power, embedded and sensor-based computing, video game design, and software engineering. He serves on the editorial committees of several journals and is currently the editor-in-chief of the International Journal on Embedded Systems and Its Applications.



Prof. Ravi Bhushan Mishra is a Professor and the Head of the Computer Engineering Department at Indian Institute of Technology (Banaras Hindu University), Varanasi. He has 34 years of teaching and research experience with more than 203 research publications. He has published various books and book chapters with publishers of international repute. His research areas include Machine Translation (MT), Multi Agent System (MAS), Semantic Web Services (SWS), Medical Computing, Intelligent Tutoring Systems (ITS), E-commerce, Robotics, Learning System in Computer Education, Cognitive Computing, and Computational Bioinformatics. He has supervised 19 doctorates and more than 42 masters students. He has visited and lectured in USA, UK, and NEPAL. He has been awarded certificate of merit in India and UK. He has organized various national and international conferences in computing.

