SRH Hochschule
Heidelberg

Task Scheduler

11005899 – Arun Vijayraghavan

Guides:
Prof. Christoph Hahn

Table of Contents

Introduction:

Task Scheduler is a system which is used to create and edit an IT project. Task scheduler is also used to manage an ongoing project where a super user or manager in this case can create tasks and delegate these tasks to the employees of the project. Task scheduler is used in most of the IT organizations as a guiding tool as well as an authentic way of keeping track of the projects. Task scheduler can be implemented in any IT organization irrespective of the software methodology which has been incorporated. Task scheduler has been proven to be one of the most successful tools in recent times for realizing the milestones of a given project in definite timeline.

Without a task scheduler, the organization will have to allocate a task force in maintaining the progress of a given project. This process tends to be very inefficient and takes more time and costs more money than it is necessary. To avoid this and to automate the process the idea and concept of task scheduler was invented. In the concerned project a working prototype has been developed and has been successfully realized with test data.

The following are the features of the task scheduler prototype which has been designed:

- There are two types of users, one of them is a super user which in this case is termed as a manager and another type of user in the itinerary is a normal user which has been termed as an employee working under that manager.
- The homepage of the task scheduler consists of login and register options. Choosing the login option will divert the user to a screen where there are username and password fields. Register option even though it appears for both kinds of users but it functions exclusively for managers.
- Manager once he logs in, he/she has the ability to create users and as well as tasks. Manager is the administrator of the task scheduler application.
- The accounts of employees have to be created by the manager, once the account has been created by the managers, employees can login to the task scheduler application with their respective username and passwords.
- Once logged in, the employees can look into the projects which they are associated with. If the employees click on any of the projects which they have been associated with, they can further look into their respective tasks. Once a particular task has been completed they can update the status of their tasks by updating the status option which represents the project progress in the task scheduler application.
- One more important aspect of the task scheduler application is the presence of the start date and the end date of a project. These field can help the employees to stay on track with the delivery date of the project. In case the dates has to be updated, the employees will have to request the manager to perform this action as they do not have the rights to alter the start date and the end date of the project.

SRH Hochschule
Heidelberg

Project Description:

The implemented prototype of the Task Scheduler currently supports the following operations,

1. Creating (also registering new users)
2. Reading
3. Updating
4. Deleting, projects that are added to the application.

Task Scheduler application prototype is implemented using HTML5 and Bootstrap CSS as the front-end and the default SQL Server, i.e. SQL Server 2012 LocalDB, serves as the back-end. Server-side scripts that triggers the database requests is implemented using ASP.NET framework 4.5 leveraging MVC5 version.

The proposed application can be used in an organization for tracking the project progress and also assists the user to keep track of the delivery date for that project. Users having “Manager” role are only allowed to create projects and also register new users for the application and users having “Employee” role are allowed to view the projects that are to be completed and are able to only update the progress status of the project.

Data manipulation function:

Create, retrieve, update and delete (CRUD) refers to the four major functions implemented in the application.

The CRUD functions are the user interfaces to databases, as they permit users to create, view, modify and alter data of the database.

The commands corresponding to these operations in SQL are INSERT, SELECT, UPDATE and DELETE. INSERT adds new records, SELECT retrieves or selects existing records based on selection conditions, UPDATE modifies existing records and DELETE removes tables or records. [1]

The implemented task scheduler prototype allows organizations for improving the project management and avoid loss of time and money at early stages. This is such a generic tool that can be adopted in any company irrespective of their business line.

Project Architecture:

Model–view–controller (**MVC**)

Model–view–controller (**MVC**) is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. The central component, the *model*, consists of application data, business rules, logic and functions. A *view* can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants. The third part, the *controller*, accepts input and converts it to commands for the model or view.

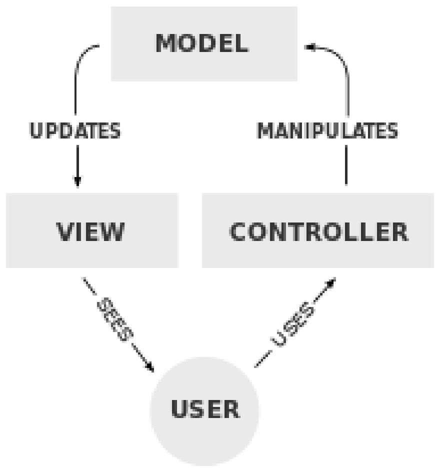


Fig. MVC Architecture

In addition to dividing the application into three kinds of components, the model–view–controller design defines the interactions between them.

- A **controller** can send commands to the model to update the model's state (e.g., editing a document). It can also send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document).
- A **model** notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. In some cases an MVC implementation might instead be "passive," so that other components must poll the model for updates rather than being notified.
- A **view** requests information from the model that it needs for generating an output representation to the user.[2]

Project Implementation:

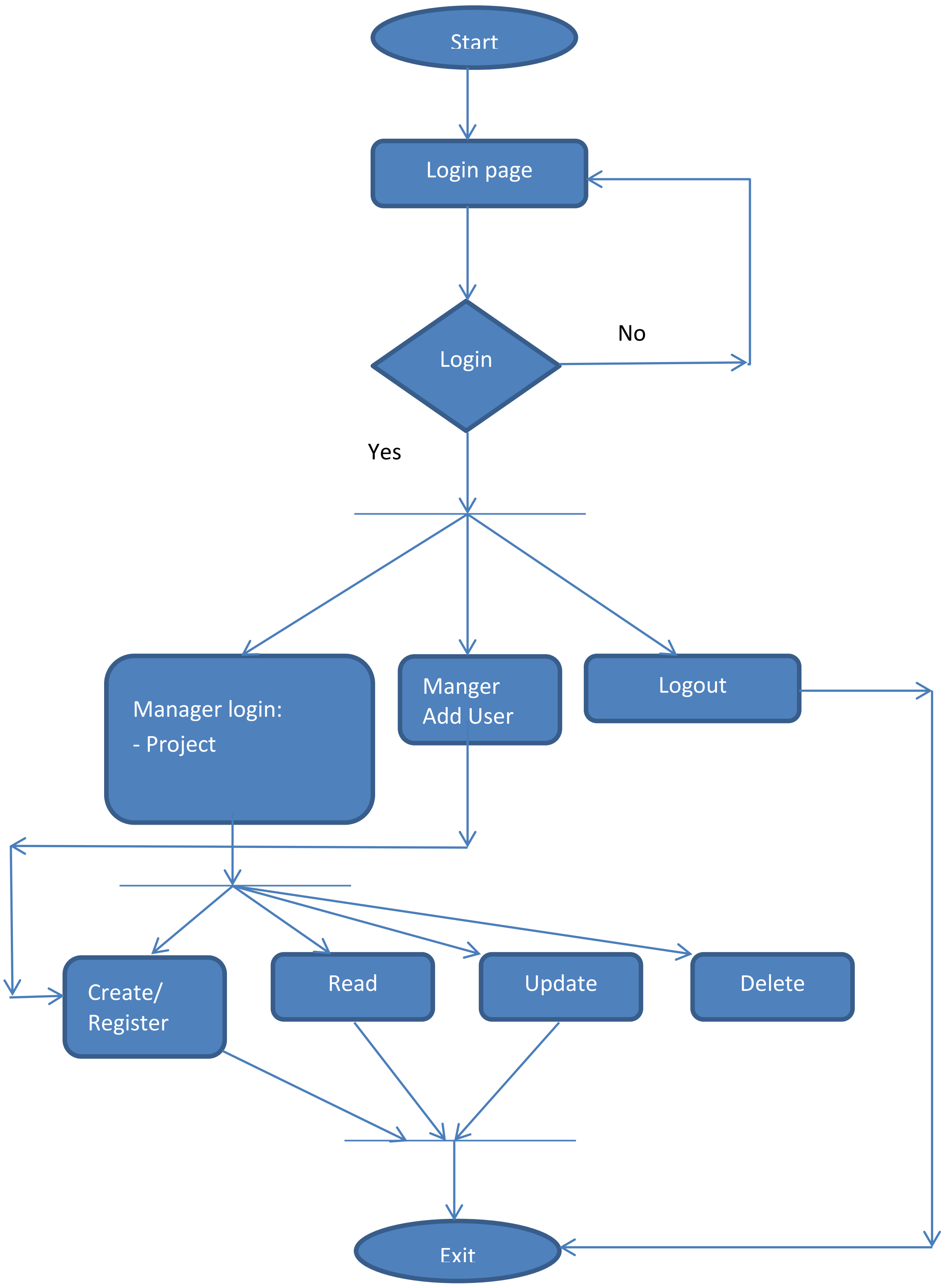


Fig. Flowchart for Manager Role

SRH Hochschule
Heidelberg

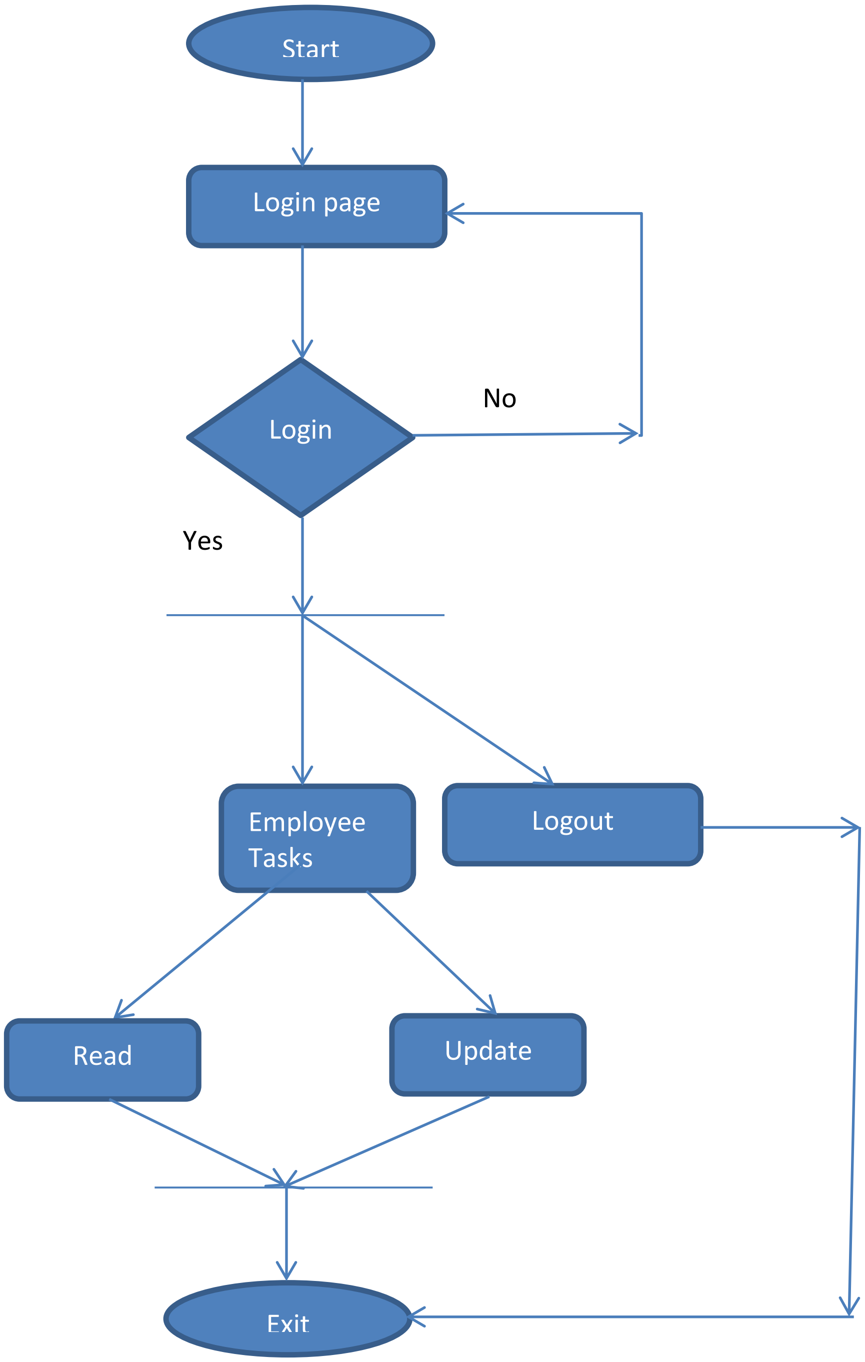


Fig. Flowchart for Employee Role

The manager has the following rights:

1. The user has to first successfully login or else the user is brought back to the login page.
2. Once the user has successfully logged in, the user has managerial access to the application.
3. The user can perform the following actions:
    a. Creating, Reading, Updating and Deleting projects.
    b. Creating new user with employee and manager roles.
    c. Logout.
4. In the Register option, the user can create username and password for other users.
5. The logout option expires the user's session and redirect to the login page.

The employee has the following rights:

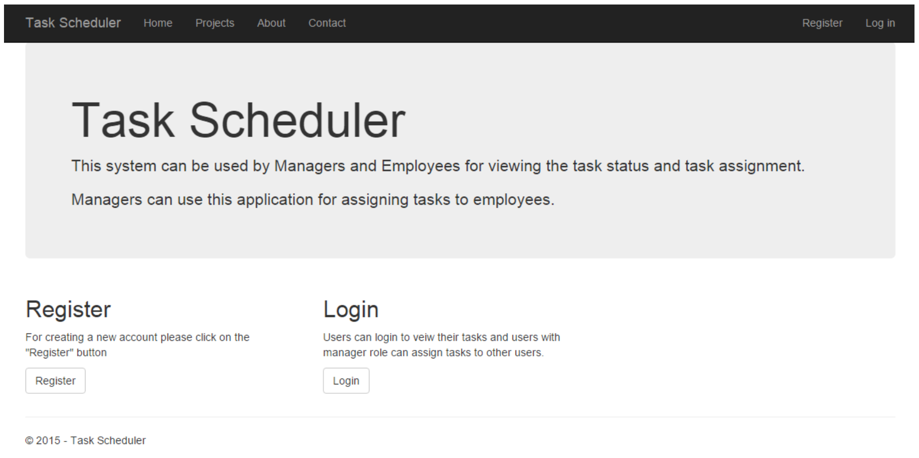
1. The user has to first successfully login or else the user is brought back to the login page.
2. Once the user has successfully logged in, the user has limited access to the application.
3. The user can perform the following actions:
    a. Update the project status.
    b. Logout.
4. The logout option expires the user's session and redirect to the login page.

The above mentioned actions are supported with CRUD operations.

Keeping the MVC pattern in mind the implementation was divided into 3 steps:

1. **Model:** The model is a part of the application that is responsive for depicting the database properties. Creation of the schema and tables was the first step taken to in order to proceed with the further implementation of the project.
2. **View:** The GUI developed using HTML5 and CSS3 forms the View for the application and creating the view was the second step for building the application. Mobile version web page rendering was accomplished with Bootstrap CSS.
3. **Controller:** ASP.NET plays the role of controller. Depending on the user's actions received through the browser, C# interacts with the database and returns the data back to the browser for the user to view.

Another aspect kept in mind during the implementation was to **maximize the reusability of the code**.



Task Scheduler Home Projects About Contact Register Log in

Task Scheduler

This system can be used by Managers and Employees for viewing the task status and task assignment.

Managers can use this application for assigning tasks to employees.

Register

For creating a new account please click on the "Register" button

Register

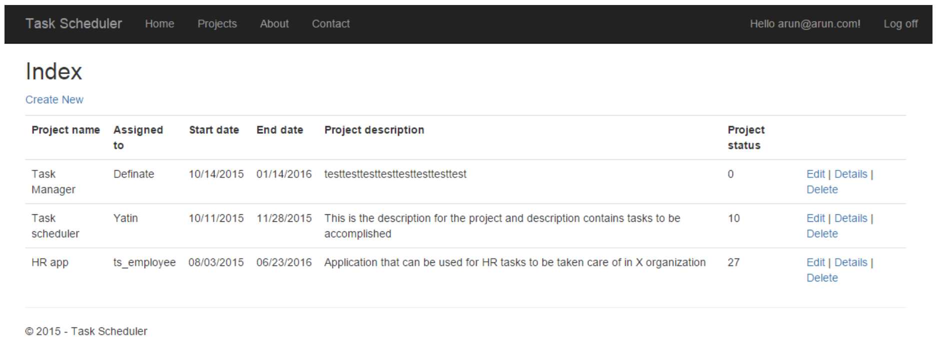
Login

Users can login to veiw their tasks and users with manager role can assign tasks to other users.

Login

© 2015 - Task Scheduler

Fig. Home page



Task Scheduler Home Projects About Contact Hello arun@arun.com! Log off

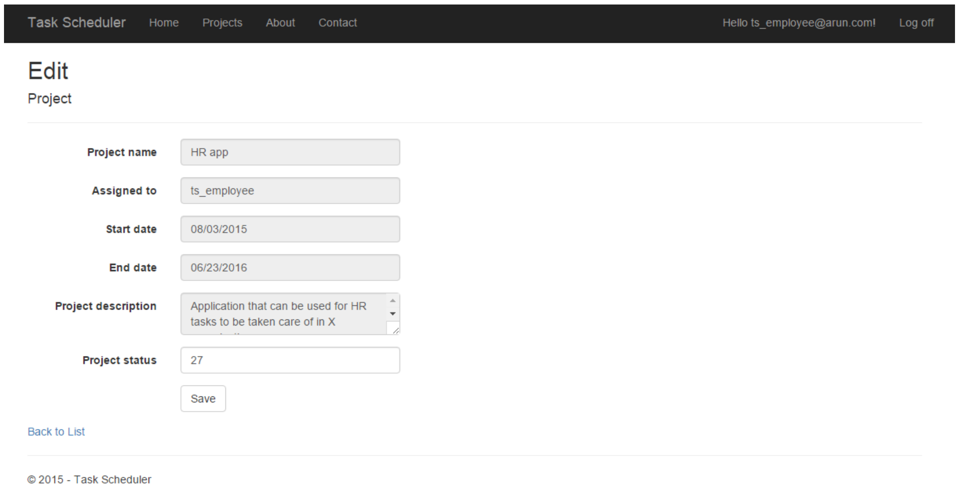
Index

Create New

Project name	Assigned to	Start date	End date	Project description	Project status	
Task Manager	Definate	10/14/2015	01/14/2016	testtesttesttesttesttesttest	0	Edit | Details | Delete
Task scheduler	Yatin	10/11/2015	11/28/2015	This is the description for the project and description contains tasks to be accomplished	10	Edit | Details | Delete
HR app	ts_employee	08/03/2015	06/23/2016	Application that can be used for HR tasks to be taken care of in X organization	27	Edit | Details | Delete

© 2015 - Task Scheduler

Fig. Projects View



Task Scheduler Home Projects About Contact Hello ts_employee@arun.com! Log off

Edit

Project

Project name	HR app
Assigned to	ts_employee
Start date	08/03/2015
End date	06/23/2016
Project description	Application that can be used for HR tasks to be taken care of in X
Project status	27

Save

Back to List

© 2015 - Task Scheduler

Fig. Projects edit view for Employee (only status update is allowed)

Important code snippets:

Some of the important code snippets which are crucial in the project are given below:

```
public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id =
UrlParameter.Optional }
            );
        }
```

The above code snippet creates a routing to navigate to a particular page. If the URL exists then user will be granted access to the pages in the application.

Steps to implement the Task Manger:

Requirements:

1. Visual Studio Express for Web installation.
2. SQL Server is usually in-build in the above mentioned installation package, if not then installation of an SQL community server is required.
3. The Web Config file should contain the connection string to the database as follows

```
<connectionStrings>
    <add name="DefaultConnection" connectionString="Data
    Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\aspnet-TaskScheduler-
    20151010030045.mdf;Initial Catalog=aspnet-TaskScheduler-
    20151010030045;Integrated Security=True" providerName="System.Data.SqlClient"
    />
</connectionStrings>
```

4. A folder named APP_DATA contains the database files which can be include into project.
5. The Routes that the application should follow are specified in the RouterConfig.cs file.
6. Whenever a controller is created, the corresponding View file inside 'View' folder is also created. The Models lie in a separate folder. This helps in code reusability and systematic application development.
7. Managerial access to the task manager can be done using the following credential:
    a. Username: arun@arun.com
    b. Password: Jay@150590

Conclusion:

The proposed Task Manager is enabled for serving the user with creation, editing and deletion of the Projects. Also security features are applied to the application by distinguishing manager and employee roles

The system can be expended to support logs of each task performed by the employees for reporting purposes and this information can be used by managers for performance reviews.

SRH Hochschule
Heidelberg

References:

Fig. MVC Architecture - http://upload.wikimedia.org/wikipedia/commons/thumb/a/a0/MVC-Process.svg/200px-MVC-Process.svg.png
[1] - http://www.techopedia.com/definition/25949/create-retrieve-update-and-delete-crud
[2] - http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller