System Specification and Design

EstiMate Web Application

Contents

Glossary of Terms

Bucket - a digital representation of a bucket with a percentage label that the user can sort real life events into (if the user is accurate, 75% of the events in the “75%” bucket will occur).

Completed Task- a user task that has met all its criteria and is submitted for reporting.

Dropped Task- a user task that the user had identified as an unachievable project. It is not submitted for reporting.

Prediction Algorithm - a set of functions that use past user data to predict how long a user will take to finish an assignment given their predictions and start time.

Probability Calibration - a module within the product where the user can make estimates on how likely real life events are to occur and then get feedback on how accurate their predictions were.

Real Life (Estimation) Event - any well-known action or occasion whose outcome can be categorized as having occurred or not having occurred.

Report - a document provided by the system containing data on the user’s predicted and actual times taken for tasks, and their accuracy in predicting time taken on tasks and event probability.

Researcher - an interest party employed by the owner of the application to study trends and anomalies in user data.

System Moderator - a person who will be given authorization to approve or deny submissions for real life events, and to confirm the outcome of approved events.

User Task - an assignment or project the user has to complete that is submitted for recording.

Product Description

The EstiMate Web Application will help users plan, record, and review the tasks they complete. A user will be able to log into the app, create and view tasks they plan on completing, and record and then view the results.

The app will attempt to predict the outcome of events input by the user, and whether the user will complete them within the predicted number of hours. They will also have the opportunity to practice their estimation abilities by betting on certain real life events. These events will be approved by system admins.

After a user has completed either a few tasks or estimation events, they will be able to view reports containing various information about their performance.

Assumptions

1. The User is entering data correctly and honestly
    1. The User is not entering data to manipulate the prediction algorithm into giving the User a better prediction score
2. The User takes responsibility to record their login information
    1. The User does not need to reset their login credentials more than once (1) per day
    2. The User does not expect their password to be reset immediately upon request
    3. The User will not reveal their login information to others
3. The Admin will not abuse their knowledge of user login information
    1. The Admin will not login to User accounts

Requirements

1. User Account

1.1. The user shall be able to create an account

1.1.2. The user shall provide a username

1.1.2.1. The username shall be unique to each user

1.1.3. The user shall provide a password

1.1.3.2. The password shall be at least 10 symbols long

1.1.3.3. The system shall not store the plaintext of the users password, only a hash of the users password

1.2. The user shall be able to log into their account using their username and password

1.2.2. The system shall send the user to the main menu if the credentials are in the database

1.2.3. The system shall prompt the user to try again if the credentials don't match

1.2.4. The system shall be able to log the user out of their account upon request

2. Tasks

2.1. The user shall be able to view all ongoing tasks

2.2. The user shall be able to create a new task

2.2.2. The new task shall contain the name of the task

2.2.3. The new task shall contain a start date

2.2.4. The new task shall contain an optional end date

2.2.4.4. The optional end date shall not be changed after creation

2.2.4.5. Tasks that do not have end dates shall not be included in calculations

2.2.5. The new task shall contain an estimation time in hours

2.2.6. The new task shall have the option to start with a total number of hours logged

2.3. The user shall be able to edit a task

2.3.2. The user shall be able to edit the task name

2.4. The user shall be able to add hours logged

2.4.2. The system shall take the added hours and add that to total hours worked

2.4.3. The system shall take current date and time when being added

2.4.4. The user shall be able to quickly add hours in increments with buttons

2.4.5. The user shall be able to add custom hours to a task

2.5. The user shall be able to end a task

2.5.2. The user shall be able to mark a task as a completed task

2.5.3. The user shall be able to mark a task as a dropped task

2.5.4. The user shall be able to view ended tasks

2.6. The Estimate System shall have a prediction algorithm

2.7. The prediction algorithm shall incrementally devalue old data (optional)

2.8. The system shall measure date and time to the nearest minute.

2.9. The system may tell the user whether they tend to be over or under confident in their predictions, and by how much.

2.10. The system may allow users to mark a task as 'will not complete', and the task will not be counted in any further calculations

3. Probability Calibration

3.1. Users should be able to guess estimates on local and world events

3.1.2. Users will be able to select buckets for estimation options

3.2. Users should be able to choose a category to see available events for that category

3.3. Users shall be able to submit suggestions of real life events for the probability calibration

3.3.2. Suggestion shall include a date by which the result of the event will be known

3.4. Users' accuracy in estimating shall be included in reports generated by the system

3.5. Probability calibration shall contain an estimation slider that begins at 50% and represents the likeliness the user believes the event will happen

3.5.2. The slider shall increment by 10%

3.6. Probability calibration shall contain a toggle button to indicate whether they believe the event will happen or not

4. Reports

4.1. Reports shall be automatically generated every month

4.1.2. Users should be able to set when reports are generated

4.1.3. Users should be able to manually generate reports

4.2. User should have option to generate report between two dates

4.2.2. The default report start & end dates shall be the generation date of the previous report (start) & the current timestamp (end)

4.3. The system shall show reports within the web app, and not send external emails

4.4. Reports shall contain information on the user's activity

4.4.2. Reports shall be generated on the user's logged task hours in relation to the date they occurred

4.4.3. Reports shall be generated on the user's estimations

5. Admin

5.1. Admin shall be able to edit suggestion question provided by a user

5.1.2. Admin shall be able to make edits to suggested exercises for language and readability

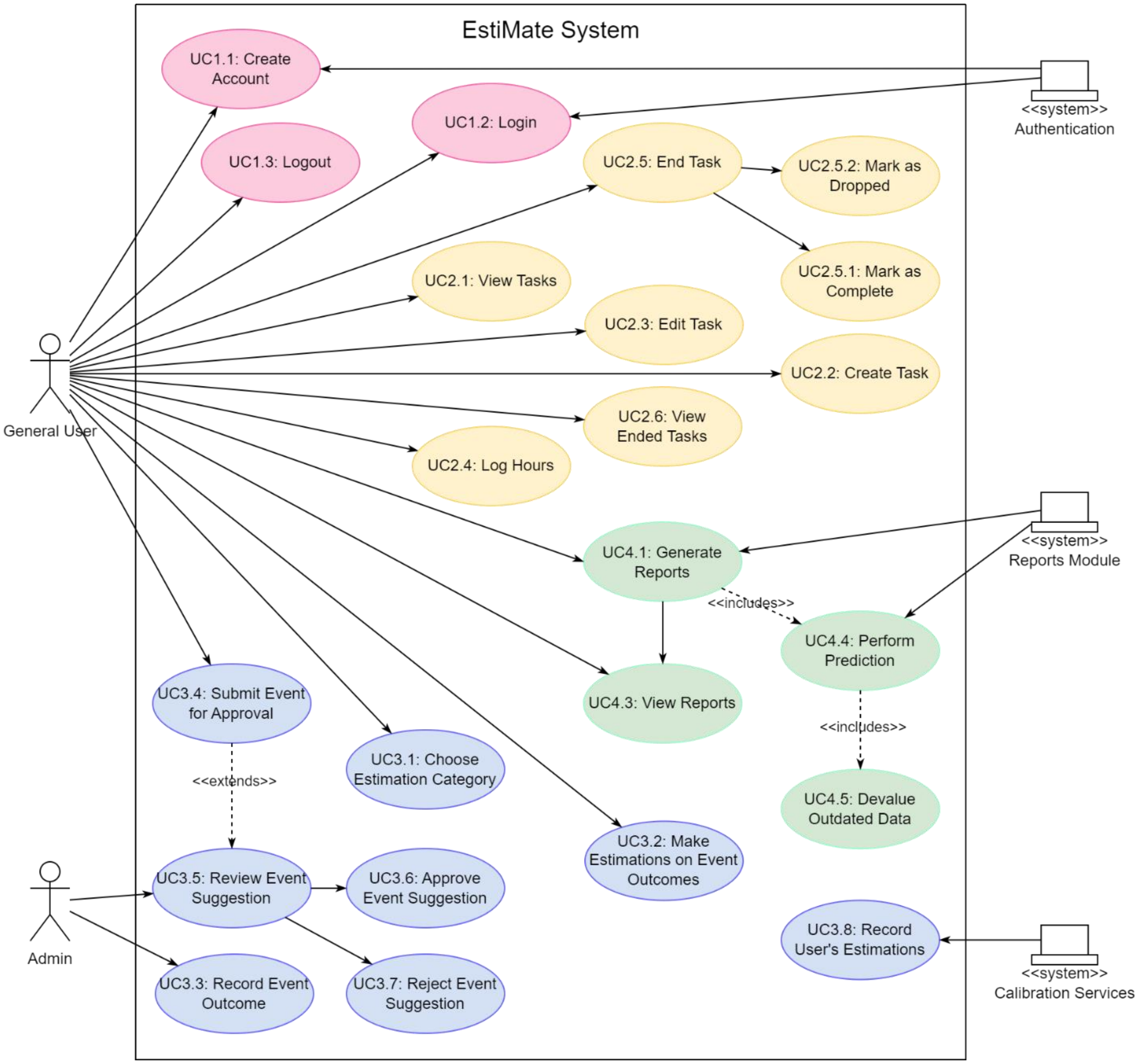
5.1.3. Admin shall be able to create a pool of questions suggested by users

5.2. Admin shall be able mark off outcomes of events

5.3. Admins will have the ability to approve or reject events submitted by users

5.4. Admins shall be able to report the outcome of real-life events

Use Case Diagram



<u>Use Case categories</u>

Account

Tasks

Estimation Calibration

Reports

Use Case Specifications

Name:	Create Account
Requirement(s)	1.1
Use Case #	1.1
Actors	User, System
Trigger	User selects 'Create Account'
Pre-Conditions	User has opened app and has a connection to the application server
Post-Conditions	User is navigated to the Login page
Basic Flow	1) The User selects Create Account 2) The System prompts the user for a username and password 3) The User enters a desired username and password 4) The System checks if the username is available 5) The System records the username and password hash 6) The System redirects the User to the login page
Alternate Flows	4a. The System notifies the User that the username is taken

Name:	Login
Requirement(s)	1.1
Use Case #	1.2
Actors	User, System
Trigger	User opens the Estimate Web App
Pre-Conditions	User has an Estimate Account and a connection to the application server
Post-Conditions	User is navigated to the dashboard page
Basic Flow	1) The System prompts the User for a username and password 2) The User enters their username and password 3) The System verifies the username and password hash match a database entry 4) The System navigates the User to the dashboard page
Alternate Flows	3a. The System determines the username and password do not match a database entry 3b. The System alerts the User of the unmatched data 3c. The System returns to step 1

Name:	Logout
Requirement(s)	1.1, 1.2
Use Case #	1.3
Actors	User, System
Trigger	User clicks “Logout” button
Pre-Conditions	User is logged in to an Estimate Account and a connected to the application server
Post-Conditions	User is logged out of the EstiMate application
Basic Flow	1) User selects Logout 2) System logs the user out of their EstiMate account

Name:	View tasks
Requirement(s)	2.1
Use Case #	2.1
Actors	User
Trigger	User clicks on “View Tasks” on the home page.
Pre-Conditions	User is logged in
Post-Conditions	User is on the task view.
Basic Flow	1) User enters the homepage while logged in. 2) User clicks on “View Tasks” 3) User can now see all tasks associated with them.

Name:	Create New Task
Requirement(s)	2.2
Use Case #	2.2
Actors	User
Trigger	User clicks on “Create new task” in the task viewer page
Pre-Conditions	User is logged in and in tasks view
Post-Conditions	User can see the new task on the tasks view.

Basic Flow	1) User enters the “Task View” page. 2) User clicks “Create new task”. 3) User enters all required information 4) User clicks “Submit”. 5) New task is added to the tasks view

Name:	Edit Task
Requirement(s)	2.3
Use Case #	2.3
Actors	User
Trigger	User clicks on a task in the task viewer page
Pre-Conditions	User is logged in and has navigated to the Task view page
Post-Conditions	User is redirected back to the tasks view
Basic Flow	1) User clicks the task they want to edit. 2) User enters and updates old information. 3) User clicks “Submit”. 4) The task view is updated with the new information. 5) User is redirected back to the tasks view.

Name:	Log Task Hours
Requirement(s)	2.4
Use Case #	2.4
Actors	Users
Trigger	User clicks the “total hours tracked” section
Pre-Conditions	User has clicked a task to edit
Post-Conditions	User is redirected back to the tasks view
Basic Flow	1) User enters the “Task View” page. 2) User clicks the task they want to edit. 3) User clicks the “total hours tracked” section 4) User enters number of hours worked on in a session 5) The system adds the new hours to the total hours 6) The task view is updated with the new information. 7) User is redirected back to the tasks view.

Name:	End Task
Requirement(s)	2.5
Use Case #	2.5
Actors	Users
Trigger	User clicks the “end task” option on task page
Pre-Conditions	User has clicked on a task
Post-Conditions	User is redirected back to the tasks view
Basic Flow	1) User enters the “Task View” page. 2) User clicks the task they want to end. 3) User clicks the “end task” button. 4) User picks the “completed” or “dropped” option from the options given 5) The system removes the task from the task view 6) User is redirected back to the tasks view

Name:	View Ended Tasks
Requirement(s)	2.5.3
Use Case #	2.6
Actors	User
Trigger	User clicks “View Ended Tasks”
Pre-Conditions	User is logged in and has ended a task (otherwise page will be blank)
Post-Conditions	User is brought to the Ended Tasks page
Basic Flow	1) User clicks “Tasks” in nav bar 2) System displays tasks page 3) User clicks “View Ended Tasks” 4) System displays Ended Tasks page (contains Completed & Dropped tasks from DB)

Name:	Choose Category
Requirement(s)	3.2
Use Case #	3.1
Actors	User

Trigger	User opens Estimation Calibration and selects a category to view
Pre-Conditions	User is logged in
Post-Conditions	User is displayed real-life events based on their chosen categories
Basic Flow	1) User enters the Estimation Calibration page 2) User clicks the “View Exercises” button for the category they wish to view 3) User is forwarded to a page containing a list of all events falling under the selected category

Name:	Make Estimations on Event Outcomes
Requirement(s)	3.1, 3.1.1
Use Case #	3.2
Actors	User
Trigger	User drags an event to a bucket
Pre-Conditions	User is logged in and chooses category to view events for
Post-Conditions	User is returned to Probability Calibration view
Basic Flow	1) User selects ‘predict outcome’ for event they wish to make an estimation about 2) User enters their predicted outcome of the event 3) User clicks “submit estimation” 4) Confirmation dialog appears to confirm their estimation was submitted

Name:	Record Event Outcome
Requirement(s)	3.5
Use Case #	3.3
Actors	Moderator
Trigger	The moderator chooses an event to report the outcome for
Pre-Conditions	At least one approved event’s outcome has been determined
Post-Conditions	The system will record the outcome of the event
Basic Flow	1) The moderator chooses an event to report the outcome for 2) The moderator marks the event as having “occurred” or “not occurred” 3) The system records the outcome of the event 4) The moderator is returned to the view of unfinished events

Name:	Submit Event for Approval
Requirement(s)	3.3
Use Case #	3.4
Actors	User
Trigger	User clicks the “Submit Event” button
Pre-Conditions	User is logged in
Post-Conditions	The event shall be submitted for moderator approval
Basic Flow	1) User clicks the “Submit Event” button 2) User is redirected to a form submission page 3) User fills out information for event name & event description 4) User clicks the “Confirm Submission” button

Name:	Review Event Suggestion
Requirement(s)	3.4
Use Case #	3.5
Actors	Moderator
Trigger	Moderator selects a user-submitted event for review
Pre-Conditions	At least one event has been submitted by a user to be reviewed
Post-Conditions	The event is approved and added to the list shown by the probability calibration or rejected and removed from the submitted events list
Basic Flow	1) The moderator selects a submitted event for review 2) The moderator reviews the information provided in the event and decides whether or not to approve it 3) The moderator clicks the “Approve Event” button 4) The event is added to the possibly shown events list and removed from the list to be reviewed
Alternative Flow	3a. The moderator clicks the “Reject Event” button 4a. The event is removed from the list to be reviewed

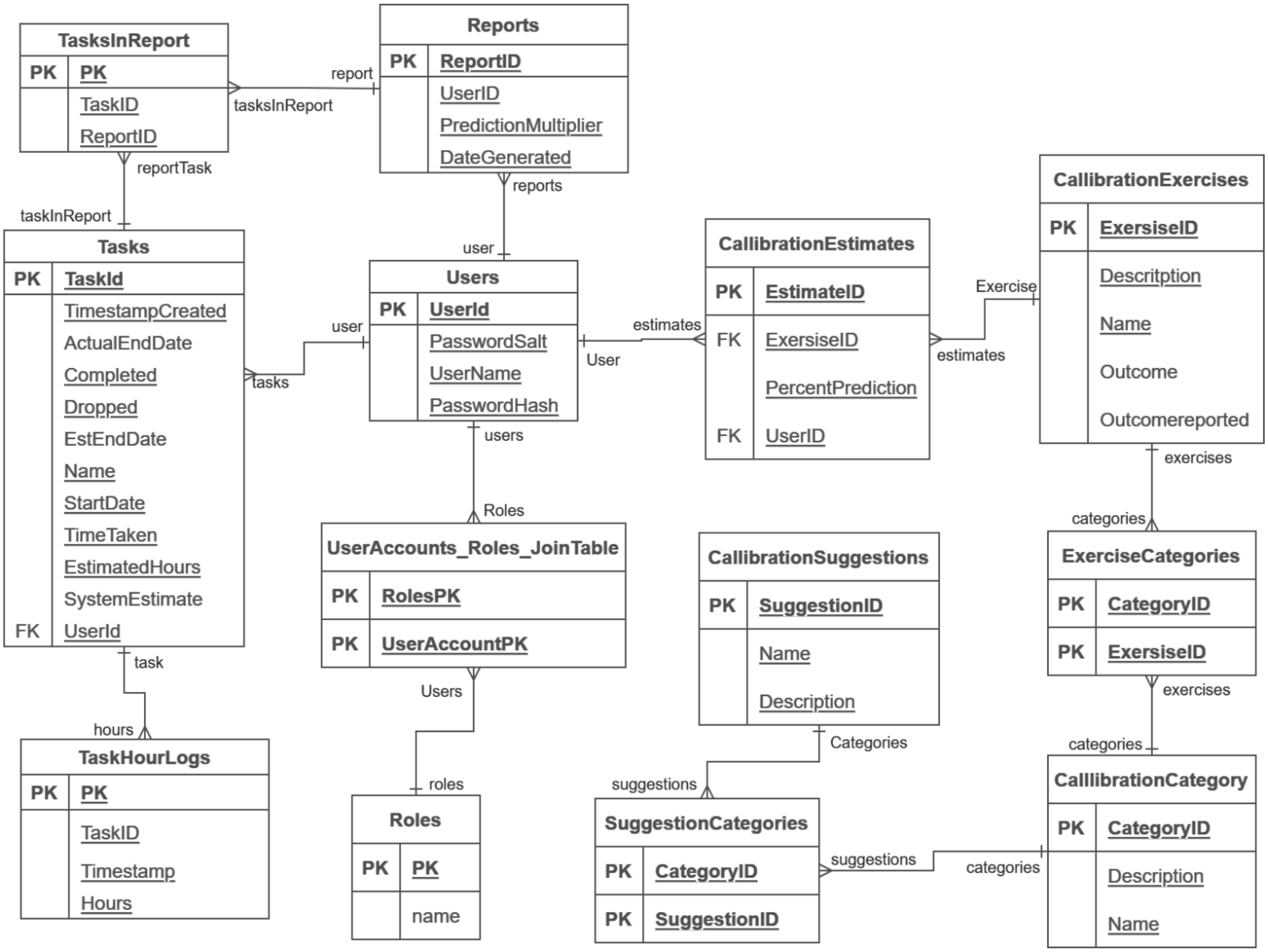
Name:	Generate Report
Requirement(s)	3.1.2
Use Case #	4.1
Actors	User
Trigger	User clicks “Generate Report”
Pre-Conditions	User is logged in and on the “View Reports” page
Post-Conditions	User generates a new report
Basic Flow	1) User enters “View Reports” page. 2) User clicks “Generate Report”. 3) Report is generated

Name:	View Report
Requirement(s)	4.1.2, 4.2.1, 4.3, 4.4
Use Case #	4.3
Actors	User, System
Trigger	User clicks “Reports”
Pre-Conditions	The user is logged in and has generated reports before (there are reports in the DB)
Post-Conditions	User is taken to the View Report page
Basic Flow	1) User clicks “Reports” in nav bar 2) System displays reports page 3) User clicks “View Report” on a report in the list 4) System queries database & retrieves report data 5) System displays full report

Name:	Perform Prediction
Requirement(s)	2.6, 2.7
Use Case #	4.4
Actors	Reports Module
Trigger	User clicks on “Generate Report”

Pre-Conditions	User has completed some tasks
Post-Conditions	System displays system estimate on how long tasks will take based on most recently generated report data.
Basic Flow	1) User enters the Reports page. 2) User clicks “Generate Report” 3) System calculates a new PredictionMultiplier to store in DB a) Old data is devalued as follows: i) Data over 2 years old has weight 0.1 ii) Data 1-2 years old has weight 0.4 iii) Data 0.5-1 year old has weight 0.7 iv) Data within last 6 months has weight 1.0 b) PredictionMultiplier = $\Sigma(actual\ time\ taken) \div (user\ estimate) \cdot weight$ for all completed tasks 4) System calculates system prediction for tasks: $(system\ estimate) = (user\ estimate) \cdot (most\ recent\ prediction\ multiplier)$

Database Design ERD


TasksInReport
PK PK
TaskID
ReportID
report
tasksInReport
reportTask
taskInReport
Reports
PK ReportID
UserID
PredictionMultiplier
DateGenerated
reports
user
Tasks
PK TaskId
TimestampCreated
ActualEndDate
Completed
Dropped
EstEndDate
Name
StartDate
TimeTaken
EstimatedHours
SystemEstimate
FK UserId
user
tasks
Users
PK UserId
PasswordSalt
UserName
PasswordHash
estimates
User
users
Roles
CallibrationEstimates
PK EstimateID
FK ExersiseID
PercentPrediction
FK UserID
Exercise
estimates
CallibrationExercises
PK ExersiseID
Descritption
Name
Outcome
Outcomereported
exercises
categories
ExerciseCategories
PK CategoryID
PK ExersiseID
exercises
categories
CalllibrationCategory
PK CategoryID
Description
Name
UserAccounts_Roles_JoinTable
PK RolesPK
PK UserAccountPK
Users
roles
Roles
PK PK
name
CallibrationSuggestions
PK SuggestionID
Name
Description
Categories
suggestions
SuggestionCategories
PK CategoryID
PK SuggestionID
suggestions
categories
task
hours
TaskHourLogs
PK PK
TaskID
Timestamp
Hours