

Mobile and Web Timecard

DESIGN DOCUMENT

sdmay18-14

Genova Technologies

Advisor: Neil Gong

Team Members

Andrew Hoelscher - Lead Tester

Christian Wessler - Lead Designer/Documentation Keeper

Cole Stephan - Lead Server Engineer

Connor McCann - Lead Android Engineer

Jason Thomas - Team Administrative Lead

Nicholas Flege - Lead iOS Engineer

Thomas Reins - Lead Web Engineer

sdmay18-14@iastate.edu

sdmay18-14.sd.ece.iastate.edu

Revised: 12/04/17-v2

Table of Contents

1 Introduction	3
1.1 Acknowledgement	3
1.2 Problem and Project Statement	3
1.3 Operational Environment	3
1.4 Intended Users and Uses	4
1.5 Assumptions and Limitations	4
1.6 Expected End Product and Deliverables	4
2 Specifications and Analysis	5
2.1 Proposed Design	5
2.2 Design Analysis	5
3 Testing and Implementation	6
3.1 Interface Specifications	6
3.2 Hardware and Software	6
3.3 Process	6
3.4 Results	6
4 Closing Material	7
4.1 Conclusion	7
4.2 References	7
4.3 Appendices	7

1 Introduction

1.1 ACKNOWLEDGEMENT

Genova Technologies has provided and will continue to provide significant guidance in the form of technical advice, tools, and financial aid. Bi-weekly meetings are held to review the progress of the senior design team, and more frequent meetings will likely be necessary with our technical lead from Genova, Tom Sidebottom, to continue answering our technical problems. Genova has set the senior design team up with accounts in the Agile tool, Rally, to allow us to document our progress in a uniform way. Genova has committed up to \$20,000 in financial support to aid in the completion of the project.

1.2 PROBLEM AND PROJECT STATEMENT

The problem we are aiming to solve with this project is Genova Technology's lack of a user-friendly and well-liked method to track time spent on projects. Genova previously had a time card which most employees enjoyed using. A change in the accounting software used by Genova left this old time card incompatible with the new accounting software. Because of this incompatibility, Genova was forced to go back to an old and disliked time tracking application. Genova hopes the senior design team can develop an iOS, Android, and web time tracking application that is both compatible with the accounting software and user-friendly.

The senior design team will solve this problem by developing time card applications for iOS, Android, and web. We will accomplish this by utilizing industry standard software development tools and by using Agile development practices with iterations every two weeks. The Agile iterations allow the team to deliver the user's highest priority needs first and facilitate adaption to change. This way we do not need to plan ahead of time for every possible need the users may have. The data for the time card will be stored, updated, and retrieved from a database hosted by Microsoft Azure. The backend codebase will be built on a Microsoft application server and coded in C#. The frontend will be coded differently depending on whether it is the iOS (Swift), Android (Java), or web app (Javascript). Both the backend and frontend will be version controlled on a private Github repository to allow for group collaboration.

1.3 OPERATIONAL ENVIRONMENT

Since this is a software project, the operating environment is fairly straightforward and we don't have to worry about any physical requirements. The iOS application will run on an Apple iOS device, the Android application will run on an Android device, and the web app will run on standard web browsers such as Chrome and Firefox.

1.4 INTENDED USERS AND USES

The end users for this product will be the employees of Genova Technologies, and potentially the employees of any companies Genova sells the software to. The mobile and web time card application must provide easy and user-friendly experiences while allowing the Genova employees to quickly and efficiently track their time spent on different projects. Additionally, it must provide a simple way for the administrators to login to the application for review and approval of timecards. If a timecard is denied, there will be an option to add a note describing why it was denied, and the timecard will go back to the employee to be changed and resubmitted. Once timecards are approved, the data must be exported to the compatible accounting software so Genova's customers can be properly billed for their time.

1.5 ASSUMPTIONS AND LIMITATIONS

Assumptions:

- Application will only be used by Genova employees and the companies Genova may potentially sell the software to; therefore the maximum number of concurrent connections will be a limited number of employees and consultants
- Users will not access time-card application from multiple platforms (on the same account) concurrently
- Multiple language support not necessary
- Voiceover support not necessary
- Application will not be used outside the United States

Limitations:

- Time-card application will have minimal feature parity with old system
- Time-card application must run natively on iOS and Android operating systems
- Server costs will be covered by client

1.6 EXPECTED END PRODUCT AND DELIVERABLES

The end product will be iOS, Android, and web applications that store, update, and retrieve information from a database hosted by Microsoft Azure. These applications will provide easy, user-friendly ways for Genova's employees to track their time spent on different projects. Once a timecard has been completed, the administrator will review it and have the option to approve or deny it. Once approved, the timecard data will be sent to the accounting software so that the customer/client can be properly billed.

The mobile and web timecard will be completed by the end of April 2018.

2 Specifications and Analysis

2.1 DESIGN SPECIFICATIONS

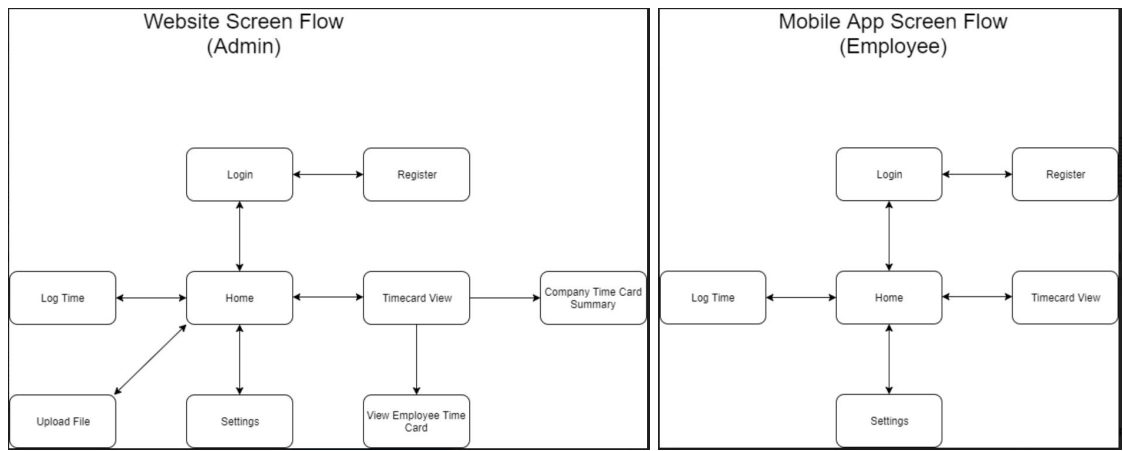
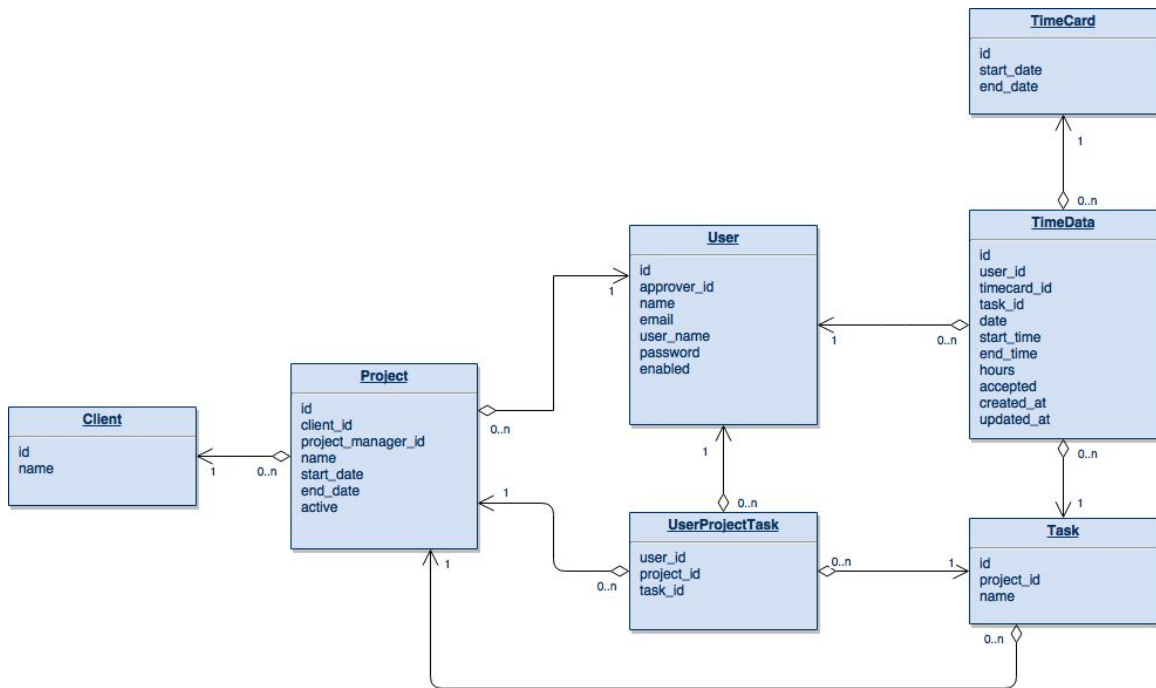
- Mobile Timecard app will function on iOS and Android devices
- Web version will support Internet Explorer, Edge, Chrome, Firefox, and Safari browsers
- Timecard backend interacts with Dynamics, Genova's accounting system
- Web and Mobile Specifications
 - User can log in with their Genova email address and password
 - New users can create accounts and passwords with their Genova email addresses
 - Mobile devices will automatically store user credentials upon first login if desired
 - Users can create a new time card, limited to one per pay period
 - Users can record time worked for a specific client, project, and task
 - Users can record time in both block and start/end time formats
 - Users can complete and submit timecard to manager for approval
 - Users can view reports of previously submitted timecards (year-to-date or last five weeks, whichever is greater)
 - Users can log out of timecard
- Web-specific Specifications
 - Managers/Reviews can review and approve their direct reports' timecards
 - Admin can view summaries of all users in timecard system
 - Admin can create timecard for another user
 - Admin can edit/complete timecard for another user
 - Admin can submit timecards for given pay period to Dynamics accounting system

2.2 PROPOSED DESIGN

The web and mobile timecard is implemented using a client-server application architecture. The application server provides the clients (Web, iOS, Android) with the data through a REST API. The application server communicates with the database to retrieve the data requested from the clients. The server is implemented with a Microsoft application server deployed on Microsoft Azure and a Microsoft SQL server which is also hosted on Azure. The clients make requests to the server for specific data using HTTP requests following the REST protocol. The data is returned to the clients in the form of JSON which is then parsed into a useable form and presented to the end user.

The database was designed following the requirements and specifications provided to us by Genova. The TimeData table is used to store a time entry for a specific user, task, and

timecard. The TimeCard table is used to track the beginning and end dates for a specific pay period. The UserProjectTask table is used to manage the relationships between the three respective tables. Users can be assigned to multiple projects and each project has a number of tasks related to it, but there are some tasks on various projects that should not be available to each user. The User, Client, Project, and Task tables are used to store various properties for each. The diagram below shows fields and relationships between the tables.



These diagrams outline the screen flows for the two variations of the client apps. The mobile app flow is on the right and the web app flow on the left. Both the mobile and web apps will have features allowing the users to create timecards, log times for a specific client, project, and task, and view previously submitted timecards. The web app will have some more advanced features for managers, who are reviewing the timecards submitted

by their direct reports, as well as admins, who will be able to see overall summary reports of all employees in the company and export timecard data to their accounting systems.

2.3 DESIGN ANALYSIS

There exist alternatives to this design are to use a Linux server and along with an SQL or NoSQL database. The server and database could also be hosted with another cloud platform such as Amazon Web Services. Technologies exist to create mobile apps in a single programming language that can then be compiled into separate iOS and Android applications. A specific requirement from Genova was to build native mobile apps, and this prevented us from implementing such technologies. We have also found through previous experience that these technologies can be very hard to debug. Members on the team also have previous experience with native iOS and Android development, so we are confident in being able to build to separate, native mobile applications.

The advantage of using a Microsoft stack for the development of this app is its integration with other Genova systems. Genova runs Microsoft software to manage their employees and finances. Since we are using a Microsoft stack, we determined using Microsoft Azure for our cloud service is the most advantageous. It has optimization for .NET web applications and integrations with Visual Studio that make deployment simple and straightforward.

We also needed to take into account that iOS development requires a Mac computer. A few team members have Macs, so they will be able to do a majority of the iOS development and there are resources available to use around campus (rentals or computer labs) in the event we need access to more Mac computers. Also, while Visual Studio was recently made available on Mac by Microsoft, many of the most useful features are only available on Windows. We are able to avoid this issue by using a virtual machine running Windows and Visual Studio. This is also an advantage for debugging/testing the mobile applications, specifically iOS, during development. For example, a local version of the server can run on a windows vm on while the iOS app is running in a simulator on macOS. The iOS app can then make local API calls instead of having to deploy the server to Azure after any change is made in order to test it with the mobile applications..

3 Testing and Implementation

3.1 INTERFACE SPECIFICATIONS

For database testing, we will be using test driven development. We will develop the tests to pull and push data to a dummy data set where the results will be evaluated for accuracy.

3.2 HARDWARE AND SOFTWARE

Hardware

The Genova time card will be deployable on any iOS or Android device, accessible from any browser, and will be hosted on a cloud-based server.

Software

We are using Rally to log user stories and use cases for the Genova time card. Developing based on these use cases will ensure functionality requirements are met. Rally will therefore serve as our center of communication for completing high-level development and testing.

Postman is being used to test the API by making various REST calls to the API endpoints. Postman is a GUI application that supports all aspects of API development and is available on all platforms. It allows for automated testing of endpoints, monitoring of API responsiveness and uptime, and mocking request responses. The ability to mock requests is very useful as it allows the client apps be easily tested with known mock data.

We will be using a full Microsoft stack and hosting the application with Microsoft Azure using MS-SQL support. Testing will be done with mock data, and evaluated by Genova for accuracy. If time permits, automated testing will be developed to ensure a seamless transition between a Genova-specific time card, and a marketable commercial time card.

The Android application will be tested dynamically using JUnit testing and generated data. Implementing the tests dynamically allows easy introduction of new test cases as the needs of the application grow with each refactor. There are third-party test suites for Android development that may be introduced as a required tool by Genova at a later date.

The web application will have multiple parts to test, and will require different languages to ensure full functionality. Selenium, a browser automation tool, in conjunction with JUnit web application testing will cover a significant amount of the application's use cases.

3.3 PROCESS

There has been no testing yet due to no functional parts of the application being created at this point in time. Everything we have done until now has been theoretical and meant to improve the team members understanding and knowledge of the project and the process.

Next semester as we build our apps and backend, we will have opportunities to test based on both functional and nonfunctional requirements. For functional portions, such as the database, web processes, and Android/iOS/Web apps, we will use test-driven development. For nonfunctional areas, we will need to work side by side with Genova. For usability, we will need to get feedback from Genova and listen to what can be improved

upon. For scalability, we will consider other companies and use their cases that exist outside of Genova so that the timecard can reach larger audiences.

4 Closing Material

4.1 CONCLUSION

With Rally set up and filled with user stories from the beginning of the semester, ongoing discussions on design and project outlook with Genova, and having the Database completed and hosted on the Azure server, we will be ready to begin functional development in the spring semester. We have created a strong base by following the Agile development method and through the guidance tactics of Genova. Their coaching through this semester has set the timeline for the project and ensures all of our deliverables have been and will be what they are looking for.

Our future steps after winter break will involve completion of core functionality on both iOS and Android device applications followed by the web application development. After functionality is completed we will return to all device applications and work on creating a standardized and easy to use design based off of Genova's original Time Card application.

4.2 REFERENCES

Android Documentation <https://developer.android.com/index.html>

Apple Documentation <https://developer.apple.com/documentation/>

Microsoft Azure <https://docs.microsoft.com/en-us/azure/>

Postman <https://www.getpostman.com>

Timesheets <https://www.timesheets.com>

Tsheets <https://www.tsheets.com>

Zenefits <https://www.zenefits.com>