

Driftr

FINAL REPORT

DAVIS NYGREN, DANIEL LEHMAN, JACOB RYAN

Table of Contents

Version Information	1
1 Executive Summary	2
2 Introduction	2
3 Problem Summary	3
4 Solution Summary	4
Front-End Summary	4
Back-End Summary	4
5 Project Challenges	5
6 Database Summary	6
Security Precautions	6
Data Integrity	6
7 Stored Procedures	7
Indexes	7
Views	8
7 Key Stakeholders	8
8 Glossary	8
9 Design Analysis	9
Advantages	9
Disadvantages	9
Appendix	10
Entity – Relationship Diagram	10
Relational Schema	11

Version Information

Version	Date	Comments
1.0	2/19/2015	Initial Draft
1.1	2/20/2015	Misc. Revisions
1.2	2/20/2015	Diagrams Updated

1 Executive Summary

This document describes the final report beginning with a description of the initial problem that this project aims to solve. Following this introduction are the problem solution, challenges of that solution, and database design summary. After the solution is presented, the strengths and weaknesses of the design are analyzed and discussed. Included at the end of this report is an appendix containing the final entity-relationship model and the final relationship schema for the system. Also present are a glossary and a list of key stakeholders in the project.

2 Introduction

This is the final report describing the Driftr system developed by Davis Nygren, Daniel Lehman, and Jacob Ryan. The purpose of this document is to join the aspect of the project discussed in previous documents and discuss the final solution. In conjunction with an examination of the final solution, this document will provide a list of first-hand feedback from its developers describing their perspectives on the advantages and disadvantages of the solution to the initial problem.

3 Problem Summary

Underground closed-circuit safety-regulation-following professional racing has been left in the dust by the social media boom. The Driftr team saw an opportunity to capitalize on and fill this void with an event-planning application ready for social media. The web application allows users to log on, create events, add their vehicles, and join events if they meet the criteria.

The project team's vision was to create a completely free social event-planning application that could be used by real professional drivers, becoming the standard for this type of communication and a significant source of advertisement traffic.

The following is a table of features as presented in the final version of the project.

Name	Feature Summary
Available as a web application	The Driftr app will be available online for home and mobile devices
Tracks information about users and their vehicles	The system will allow users to input and edit information about themselves and their vehicles
Tracks user-created events	The system will allow users to create events with the description and time that they want
Tracks user preferences for events	Allows the event creator to add restrictions for the type of vehicles they want to allow in their event, and only lets users join if they have a vehicle that meets the criteria.
Allows users to join events	Users may join each other's events provided they have a vehicle that meets the creator's preferences for the event
Allows users to add locations	Users can add locations by searching or marking them on the locations map. Current locations are marked on the map.
Tracks which users are 'friends'	Users can add each other as 'friends' and add a relationship description

4 Solution Summary

The solution developed by the Driftr team was to implement the user interface as a web application connected to Microsoft SQL Server 2012 to store all of the user information.

Front-End Summary

The web-based user interface was created with the intention of keeping navigation simple and prioritizing fast access to all of the core user data to cater to the target market's preference for speed. The website is divided into only a few different pages, navigable by the menu bar at the top of every page. Insertion, alteration, and deletion operations for events, vehicles, and other constructs are easily accessed from their appropriate page.

When the user navigates to the Driftr website they will be presented with a log in page requesting a username and password. Registration is located via a button on the menu bar and requires a name in addition to username and password. Once logged in, the user is taken to the "dashboard" panel and presented with a summary of their vehicles, events, and friends.

The events page shows a list of existing events, and allows a user to join events they are eligible for, along with an edit button for events they have created. At the bottom of the page is a form to create a new event.

The locations page features a map with the current locations marked on it, as well as a list of the current locations, and an entry field for new events.

The cars page shows a list of the user's cars as well as an entry form for adding additional vehicles.

The friends page shows a list of the user's friends as well as an entry form for adding additional friends.

The profile page shows the current email and name and allows the user's name and password to be updated.

A button to end the session and log out of the website is also present on the menu bar.

Back-End Summary

The UI accesses and displays data from the Driftr database, which includes tables tracking information about Users, Vehicles, Events, Event Participants and Preferences, Friends, and Locations. The web interface is only allowed to access the database with predefined procedures in order to enhance security. The Driftr database is currently hosted on the Titan server of the Rose-Hulman Institute of Technology CSSE department, but will be wiped before the beginning of the spring 2015 quarter. The database is discussed in greater detail in the Database Summary section.

5 Project Challenges

- **Challenge:** Division of labor
 - **Solution:** The work on the documents, front-end, and back-end were divided fairly evenly among group members, with the exception of Jacob handling the C# API.
 - **Analysis:** Early on, this strategy was very effective. Later in the project timeline it became more difficult to split the work on the project up, so the team had to communicate very clearly what each member was currently working on.

- **Challenge:** Jacob was the only one with C# experience
 - **Solution:** When the other group members were working on the front-end and stored procedures, they communicated which stored procedures needed to be added to the C# code well in advance, to make sure they could be connected on time.
 - **Analysis:** The advance notice to Jacob of the modifications he needed to make to his portion of the code was critical to completing the milestones on schedule and worked well.

- **Challenge:** Scheduling was difficult with the three team members living in different locations
 - **Solution:** The team would meet when possible and assign each group member a portion of the project to work on individually before the next meeting.
 - **Analysis:** This approach worked well, ensuring that each member the necessary components of the project.

6 Database Summary

The design is illustrated by an entity-Relationship model and relational schema included the appendix.

Security Precautions

Access to the database is only provided to the web application by a purpose-created account with very limited privileges. It is restricted to only running stored procedures that have been created for its access. This prevents database-compromising commands from being executed by the application, even if someone with malicious intent know the structure of the database. In addition, only sanitized parameters are passed to the procedures, removing the opportunity for SQL injection attacks. One security concern that is still present is the possibility of a denial of service (DOS) attack, which was decided to be outside the scope of the project.

Data Integrity

The referential integrity constraints of the database are represented by the black arrows in the relational schema diagram, present in the appendix. Transactions that would break these constraints are not allowed to execute on the database. The domain integrity constraints of the entities in the database are described in the Security Analysis.

7 Stored Procedures

Stored Procedure Name	Procedure Description
delete_eventparticipant	Removes a user from an event based on the user email and event id
delete_friend	Removes a friend based on the user's email and the friend's email
delete_location	Removes a location based on its id
delete_preference	Removes a preference based on the event id and the field of the preference
grant_execute	Grants permission for the Driftr User to execute the stored procedures
insert_event	Adds an event
insert_eventparticipant	Adds a user as a participant in an event
insert_friend	Adds a friend relationship between two users
insert_location	Adds a location
insert_preference	Adds an event preference
insert_user	Adds a user
insert_vehicle	Add a vehicle for a user
select_all_events	Selects all events
select_all_locations	Selects all locations
select_all_users	Selects all user
select_all_vehicles	Selects all vehicles
select_event	Selects an event based on id
select_eventparticipants	Selects all participants for a given event
select_events_by_user	Selects all events that a user is participating in
select_friends_by_user	Selects all friends of a particular user email
select_location	Selects a location based on id
select_preferences	Selects the preferences for a particular event
select_user	Selects a user based on email
select_vehicles_by_user	Selects all vehicles based on a user email
update_event	Updates a particular event based on id
update_eventparticipant	Updates an event participant based on participant email and event id
update_location_description	Updates a location description
update_user	Updates a user based on email
update_vehicle	Updates a vehicles based on vehicle id and user email
vehicles_allowed	Return the number of vehicles a user owns that are allowed by the preferences for a particular event

Indexes

Other than the primary key indexes that are automatically generated for each table, only a few additional indexes were created. These were created on the "Make", "Model", and "Color" attributes in

the [Vehicles] table as these are very frequently accessed when viewing the details of an event, when it is determined if the user has a vehicle that matches the required preferences of the event.

Views

Views were not considered necessary, as all data is easily reachable on either the front-end or through SQL Server Management Studio and all queries are made through stored procedures. The stored procedures themselves are formatted to prevent direct table access and the web API that exposes these stores procedures does not reveal information about the back-end architecture.

7 Key Stakeholders

Name	Role
Sriram Mohan	Project Advisor
Davis Nygren	Project Team
Jacob Ryan	Project Team
Daniel Lehman	Project Team
Closed-Circuit Safety-Regulation-Following Professional Racers	End User Base

8 Glossary

CCSRFPR - closed-circuit safety-regulation-following professional racing

Entity – Relationship (ER) Diagram – A symbolic abstraction of a database

Relational Schema – A diagram showing the tables and the referential integrity constraints of the database

9 Design Analysis

Advantages

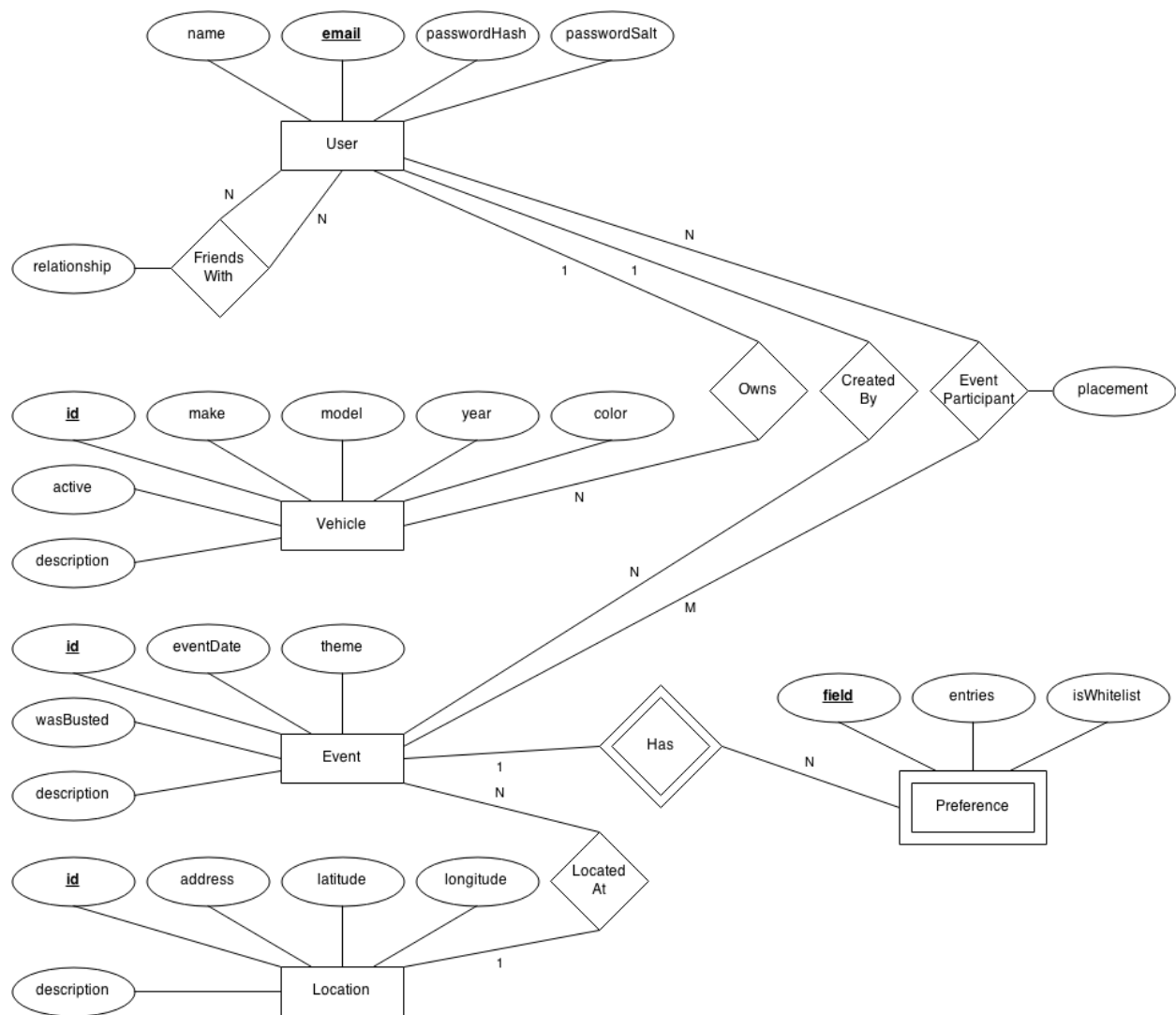
- All tables are indexed allowing for speedy query responses
- High referential integrity through rejecting all changes that would cause data to become inconsistent
- Column names are clear so users can intuitively understand their meaning with no ambiguity.
- Database has high level of security by restricting the access of the interface application to executing stored procedures
- Entirety of source code is publicly available through GitHub allowing developers to make informed changes when developing features
- Stored procedures follow a consistent naming convention, allowing for simple access
- Passwords are securely encrypted in the database. A random salt is generated for every user and is used to hash their password with industry leading security. Even in the catastrophic event of data theft, users' password would definitely remain unknown.

Disadvantages

- The C# API called by the JavaScript in the webpage isn't very clear or easily understood. Providing a publicly available documentation for the different API calls, along with examples, would alleviate this.
- The pages with sensitive data are http, not https, creating potential security issues with user data such as passwords. These could be converted to use https to encrypt the data for a production system.

Appendix

Entity – Relationship Diagram



Relational Schema

