

# DineRoulette

Personal Project – Requirements and Specifications

Github Repository – <https://github.com/tamkylet/DineRoulette-tamkylet>

Heroku Deployment – postponed

---

## Iteration: 0

DineRoulette is a web based dating application which forces both clients to be charged by a chosen restaurant of a set price through Paypal API prior to meeting up for dinner. Our target clients would benefit from the service as it significantly decreases chances of no-show which maximizes user experience. The primary goal is to ensure that our clients are amused by using our service which causes clients to remain active. This can be accomplished by yielding interesting statistics about each client based on their past experiences with DineRoulette.

Our target audience is set for young adults which will generate the most clients. Throughout large urban centres a key observation about dating scenes is that participants are predominantly young adults. As a local Canadian software company, establishing partnerships with Canadian restaurants for uses of Paypal with our service is ideal because federal and provincial laws yield less opposition towards possible features that can be implemented. The scope of this service can be scaled globally by critically analyzing international laws situated towards countries.

The broad idea of DineRoulette is to enable clients to arrange dinner dates. Each week a random sponsored restaurant with use of Google Maps API will be used for arrangement. Through extensive research, a similar dating application exists called “Dine” which self-proclaims to simplify popular dating application called “Tinder” by immediately directing both parties to select a restaurant for meetup whenever a match occurs. What differentiates DineRoulette from competitor is that our application decreases disappointment from chances of no-show, encourages habitual use with statistics, and increases familiarity with clients by storing feedback of their past dates.

DineRoulette yields a unique feature which sets a countdown timer once a client agrees to another client’s dinner request. An e-mail is sent to both participants which acts as a reminder to pay selected restaurant with Paypal. If however, countdown timer is completed before both participants paid, both participants are permanently disabled from requesting dinners with each other.

## Iteration: 1

For this iteration the following user features will be implemented:

- **Registration:** Users will be able to create an account within the website by picking a unique username and e-mail address.
- **Verification:** Verification e-mail will be sent to e-mail address used to create an account.
- **Login/Logout:** Users will be able to log into their individual accounts and logout at anytime.

- **Administrator:** Special user who can view, add, and remove users.
- **User Stories:** Attached.

## **Iteration: 2**

Local server in Cloud9 IDE blocks e-mail from being sent to deter spammers. There exist possibilities for utilizing a Google Compute Engine partner service for sending verification e-mails next iteration. Due to this constraint, once a spectator creates an account with our service, their account will automatically be verified.

For this iteration the following user features will be implemented:

- **Member List:** Each user is able to see other users on their home page.
- **Profile View:** Every user can see other user's basic personal information.
- **Restaurant of Week:** Users are able to use restaurant to arrange date.
- **Date Confirmation:** Users are able to accept or decline a date offer from other user.
- **User Stories:** Attached.

General Improvements:

- Introduction of user sessions
- Administrator user list view through categorized options
- Improved UI; especially connectedness

## **Iteration: 3**

Project acts as sample application only; this will be last iteration.

For this iteration the following user features will be implemented:

- **API:** Paypal (transaction handling), Google Maps (mapping restaurant of the day).
- **Profile View:** Every user can see other user's secondary service information.
- **Restaurant List:** Current restaurant used for dating is randomized upon user login.
- **Notification:** Notification e-mails associated to invitation process will be sent to e-mail address of matched participants.

- **Cancellation:** User with an invitation may cancel upon notification.
- **User Feedback:** User can leave feedback for other user (user's secondary service information) after date.
- **User Stories:** Attached.

General Improvements:

- Major UX and UI work with CSS and JavaScript