SCHEDULER

Ryan Tolentino
Miguel Castro

May 3, 2018

Overview

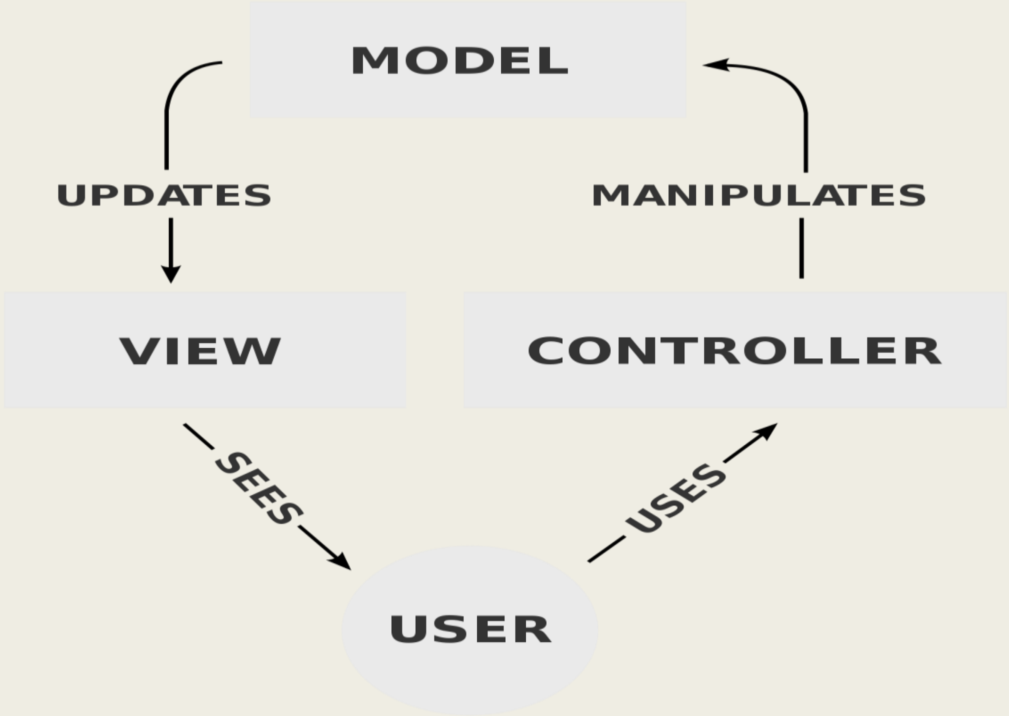
- Introduction
- Background
- Tool & Techniques
- Problem Solution
- Risk Assessment
- Struggles
- What is next

Introduction

- We have decided to create scheduler application that allows two individuals to schedule an activity. Our example in this case is scheduling a tea activity.

Background

- **Model View Controller** Web2py separates the data representation (the model) from the data presentation (the view) and also the application logic and workflow (the controller).

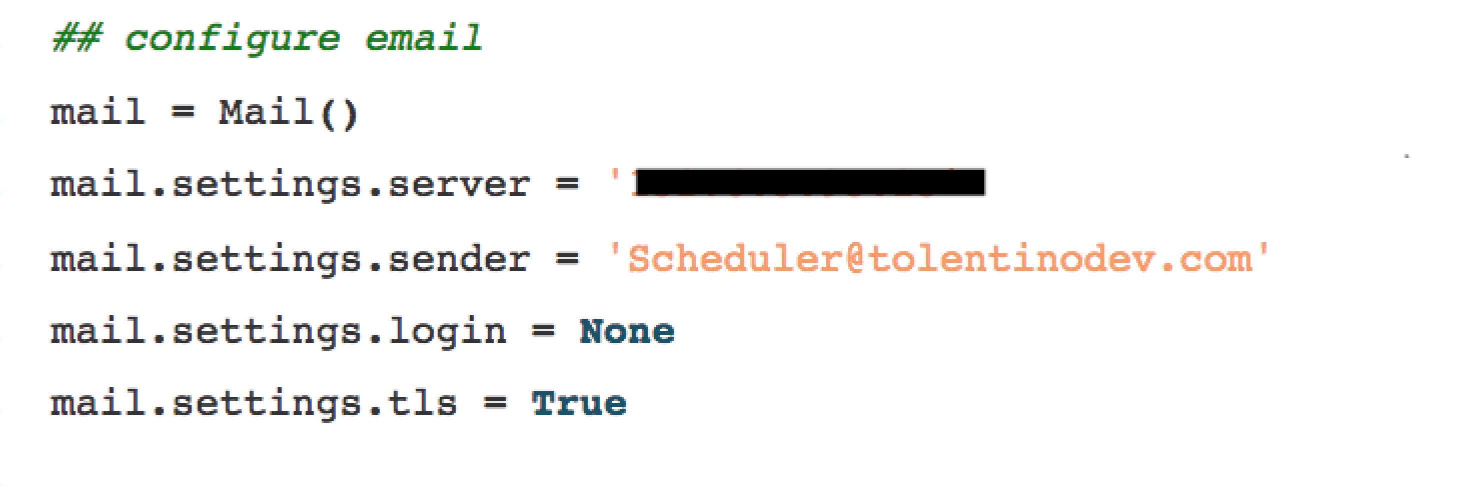


Tools and Techniques

- **Web2py**
  Web2py is a free open-source web framework for secure database web application. The application uses python and includes a database abstraction layer, therefore it knows how to generate SQL by itself.
- **SQLDB**
  A database is used for a structured set of data held in the computer. In our application we will be using an SQLDB.
- **HTML,CSS,Javascript**
  The view is the web applications visual appearance. The view is attached to the model and the controller and gets the data to present. The views utilize Hypertext Markup Language (HTML) Cascading Style Sheets (CSS) and a little bit of javascript to make the application more interactive.

Tools and Techniques

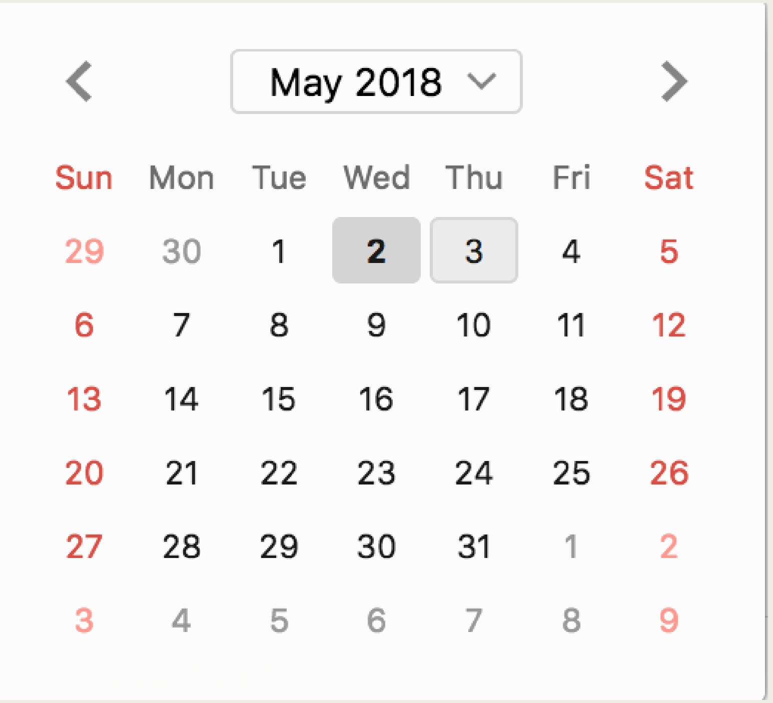
- **Mail Server/SMTP**
  This web application utilizes a single mail server to send and receive emails.



```
## configure email
mail = Mail()
mail.settings.server = '[redacted]
mail.settings.sender = 'Scheduler@tolentinodev.com'
mail.settings.login = None
mail.settings.tls = True
```

Tools and Techniques

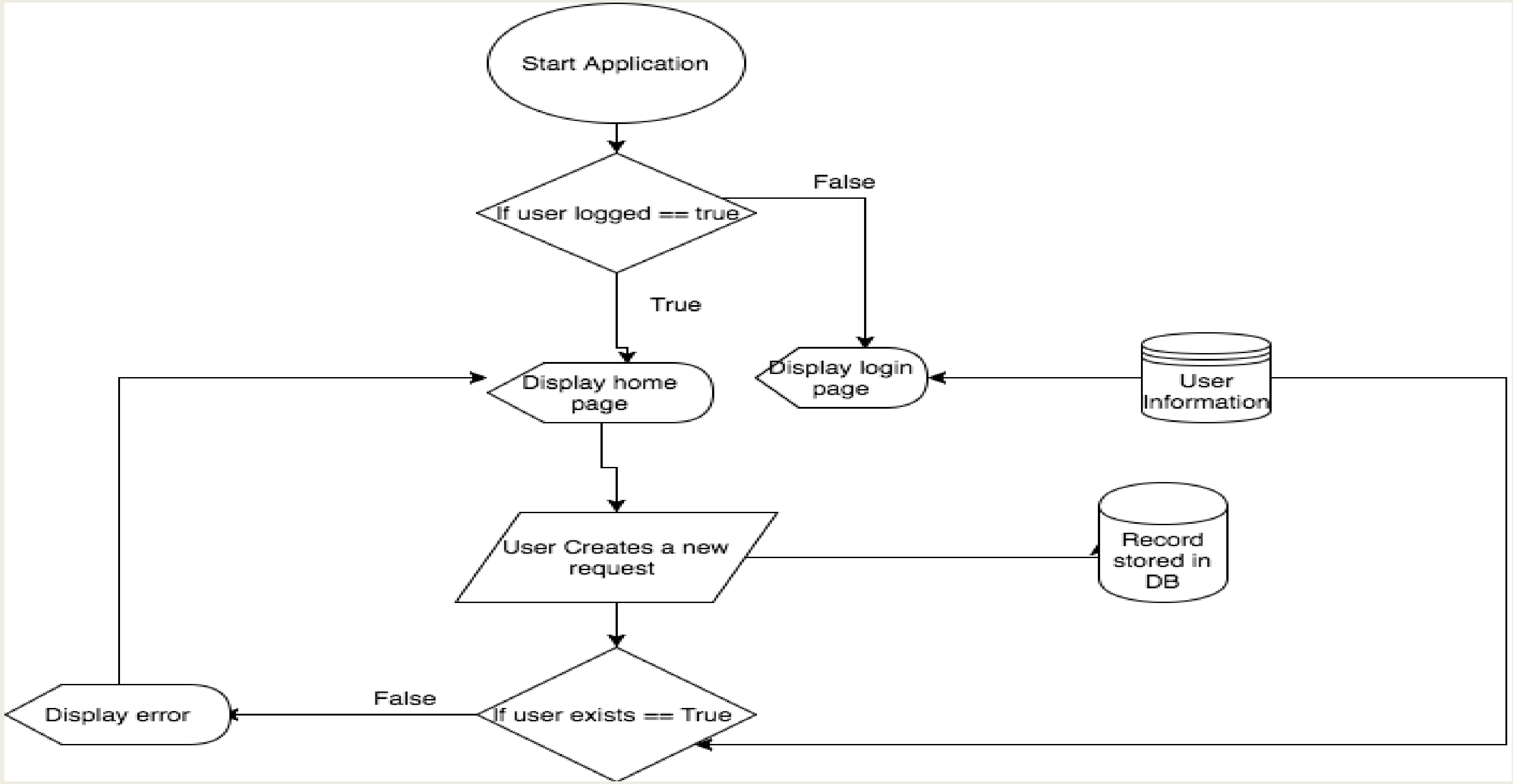
- **Calendar Application Program Interface (API)**
  Web2py allows the use of plugins and APIs. We decided to take advantage of this by utilizing a basic calendar API for ease of use.



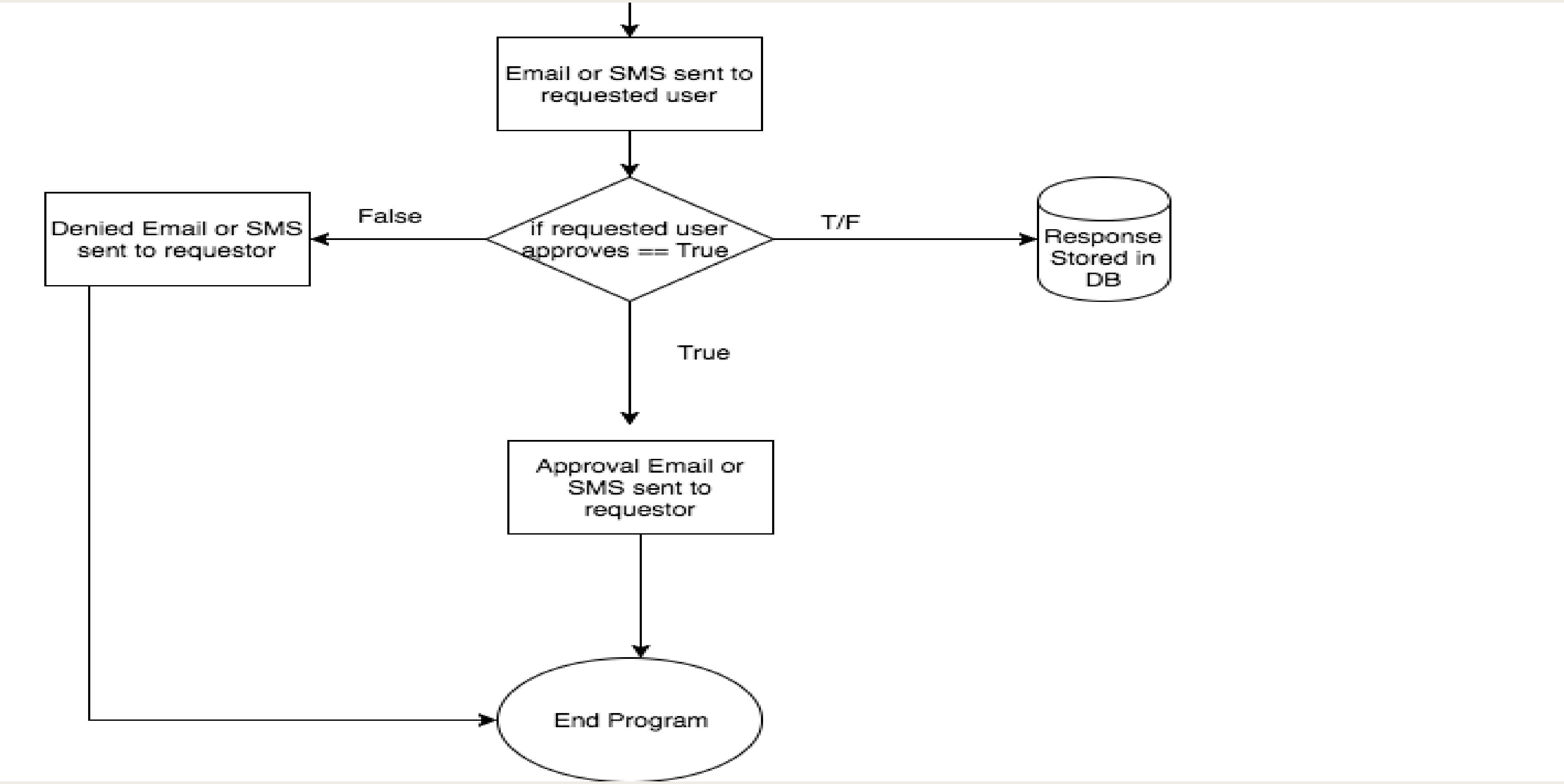
May 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Problem Solution

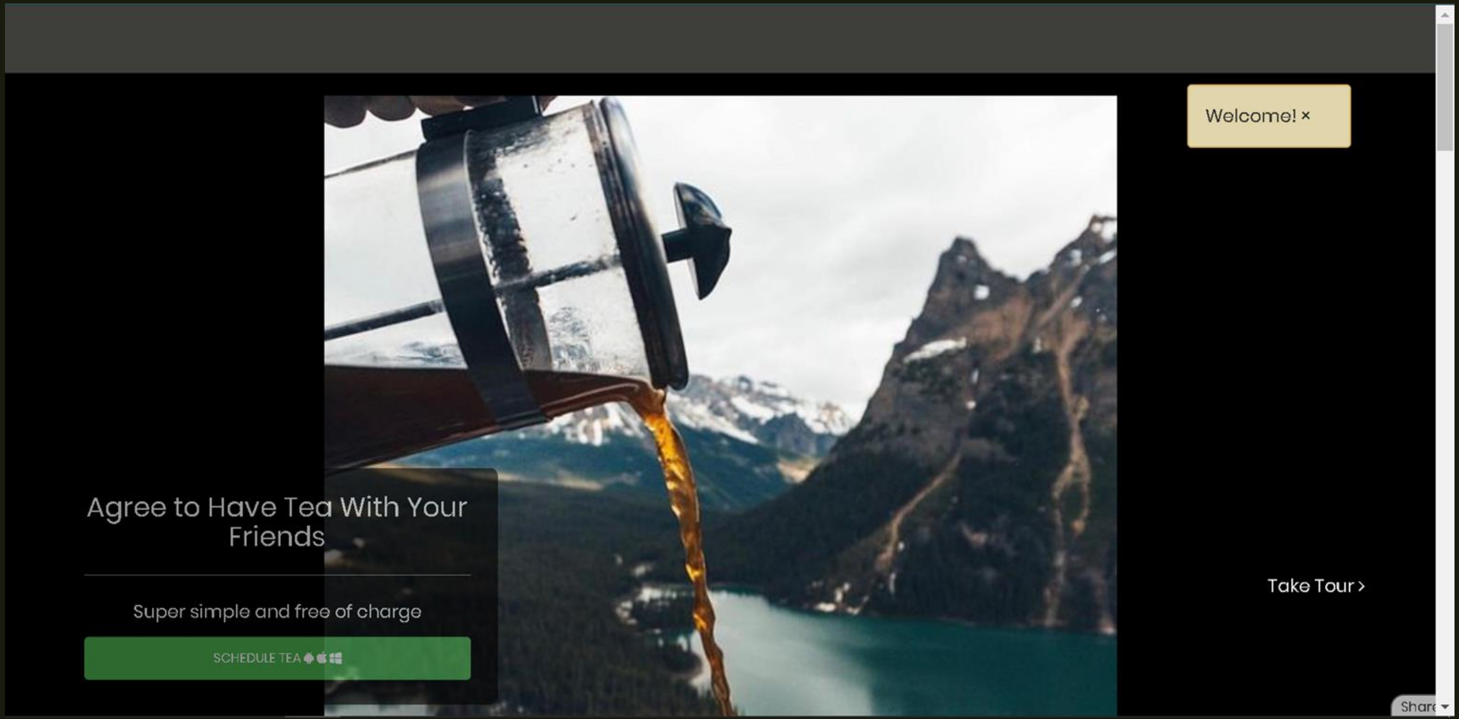


Problem Solution


Email or SMS sent to requested user
False
if requested user approves == True
T/F
Denied Email or SMS sent to requestor
Response Stored in DB
True
Approval Email or SMS sent to requestor
End Program

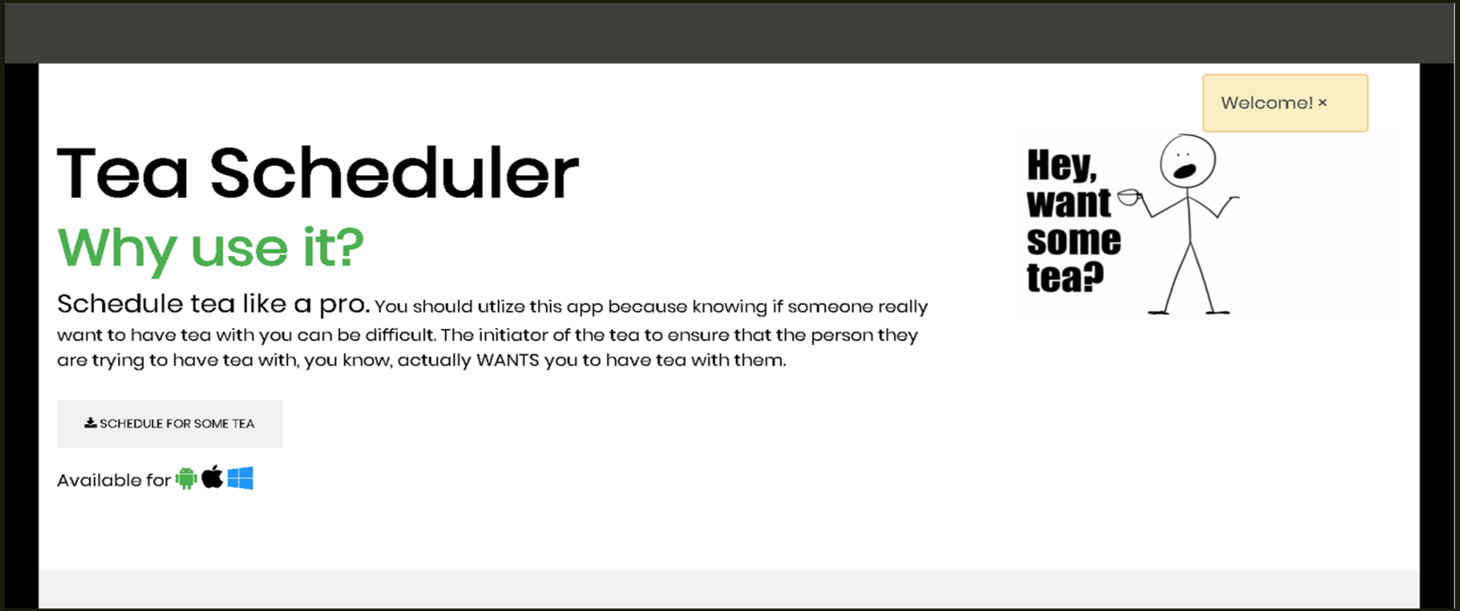
Problem Solution

- This is a prototype version of our application, only shows the view mode of the Tea App


Welcome! ×
Agree to Have Tea With Your Friends
Super simple and free of charge
SCHEDULE TEA
Take Tour ›
Share

Advertisements

- This is a prototype version of our application, only shows the view mode of the Tea App

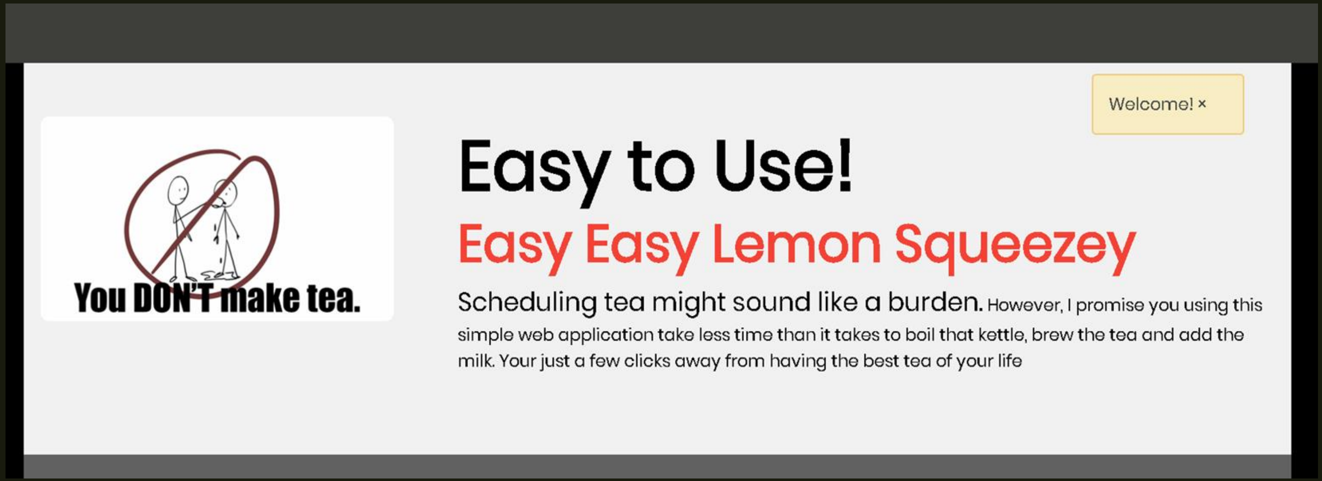


Tea Scheduler

Why use it?

Schedule tea like a pro. You should utlize this app because knowing if someone really want to have tea with you can be difficult. The initiator of the tea to ensure that the person they are trying to have tea with, you know, actually WANTS you to have tea with them.

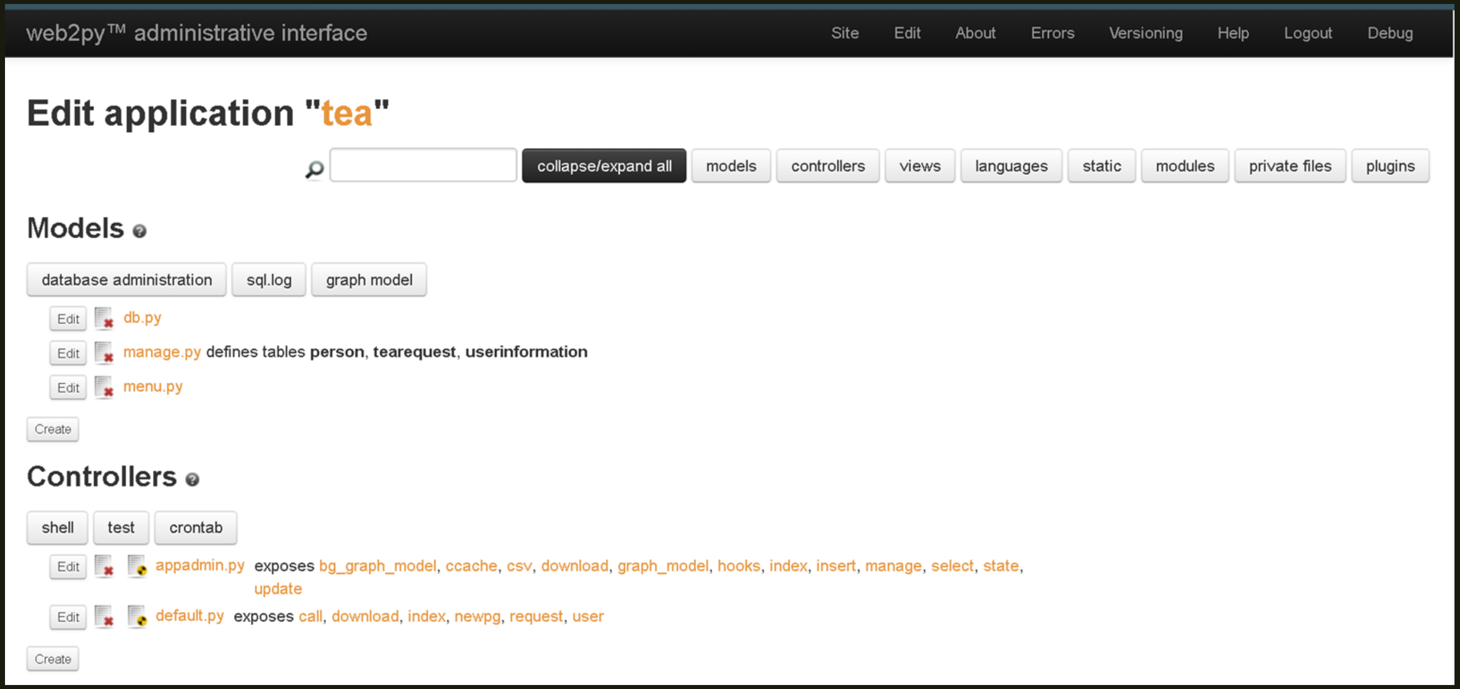
Available for



Easy to Use!

Easy Easy Lemon Squeezey

Scheduling tea might sound like a burden. However, I promise you using this simple web application take less time than it takes to boil that kettle, brew the tea and add the milk. Your just a few clicks away from having the best tea of your life



Edit application "tea"

Models

db.py

manage.py defines tables **person**, **tearequest**, **userinformation**

menu.py

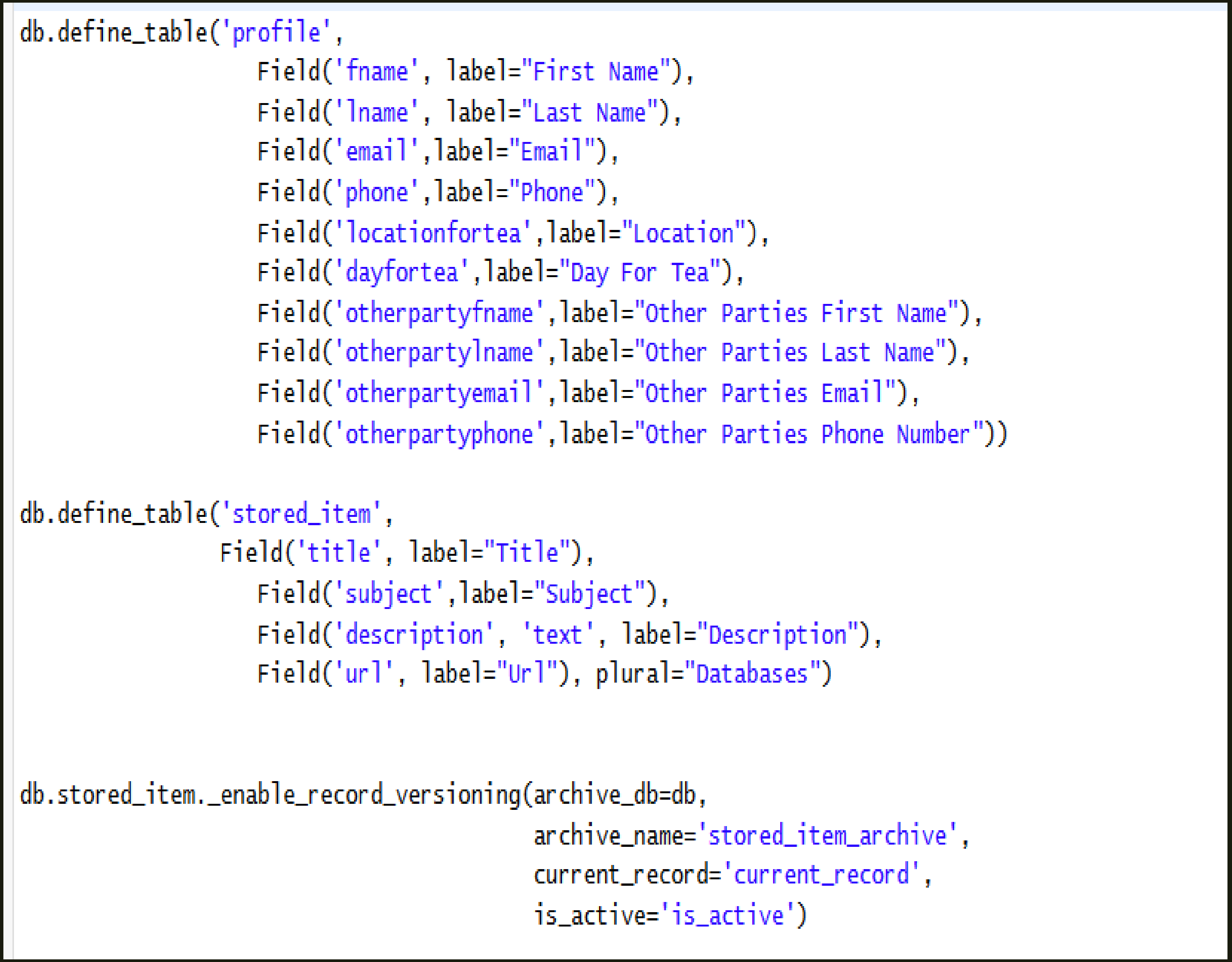
Controllers

appadmin.py exposes bg_graph_model, ccache, csv, download, graph_model, hooks, index, insert, manage, select, state, update

default.py exposes call, download, index, newpg, request, user

Web2py Admin Interface

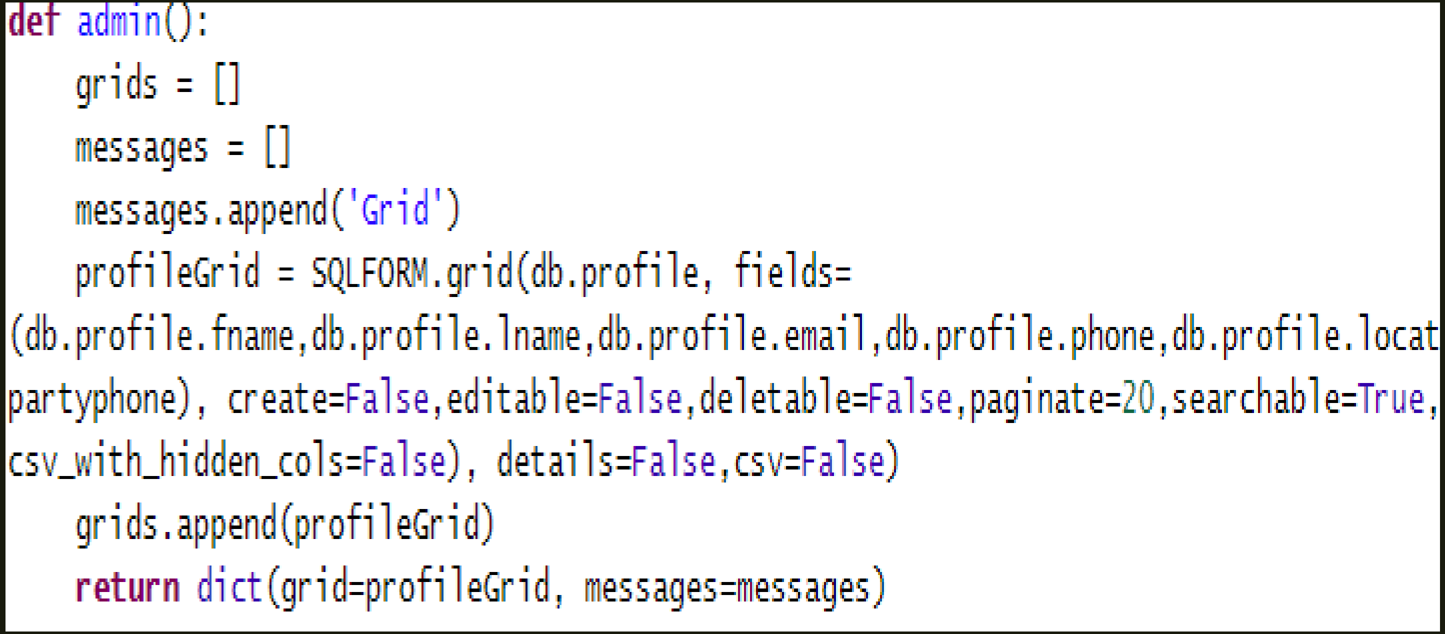
- This is how we edit the Tea App
- It uses the concept of Model View Controller Concept
- The framework uses Python, JavaScript, and HTML language



```
db.define_table('profile',
                  Field('fname', label="First Name"),
                  Field('lname', label="Last Name"),
                  Field('email',label="Email"),
                  Field('phone',label="Phone"),
                  Field('locationfortea',label="Location"),
                  Field('dayfortea',label="Day For Tea"),
                  Field('otherpartyfname',label="Other Parties First Name"),
                  Field('otherpartylname',label="Other Parties Last Name"),
                  Field('otherpartyemail',label="Other Parties Email"),
                  Field('otherpartyphone',label="Other Parties Phone Number"))

db.define_table('stored_item',
               Field('title', label="Title"),
                  Field('subject',label="Subject"),
                  Field('description', 'text', label="Description"),
                  Field('url', label="Url"), plural="Databases")



db.stored_item._enable_record_versioning(archive_db=db,
                                       archive_name='stored_item_archive',
                                       current_record='current_record',
                                       is_active='is_active')
```

Database Model

- We created a table database the defines and stored user information and tea request information
- This allows us to request both of the user's data and allow to schedule the activity



```
def admin():
    grids = []
    messages = []
    messages.append('Grid')
    profileGrid = SQLFORM.grid(db.profile, fields=
(db.profile.fname,db.profile.lname,db.profile.email,db.profile.phone,db.profile.locat
partyphone), create=False,editable=False,deletable=False,paginate=20,searchable=True,
csv_with_hidden_cols=False), details=False,csv=False)
    grids.append(profileGrid)
    return dict(grid=profileGrid, messages=messages)
```

Controller

- We created a controller that stores all the inputs from form
- This allows us to gather both parties information and to complete the request

Live Demo

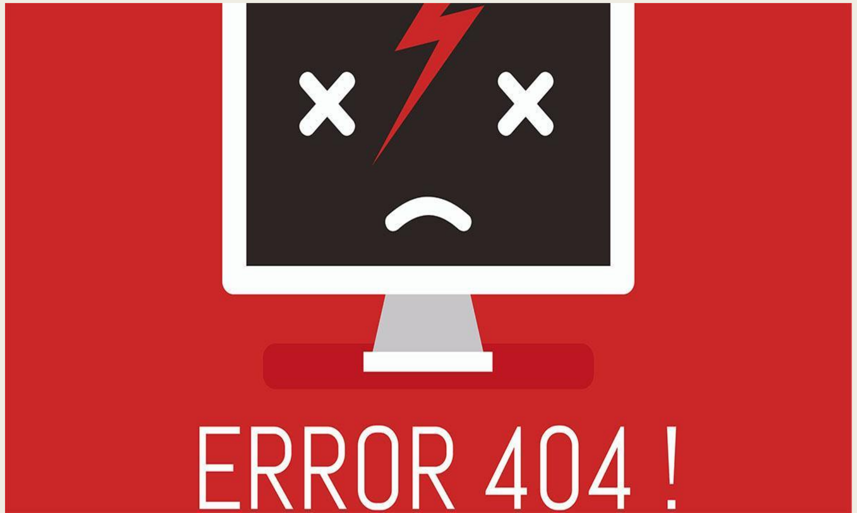
https://web.uncfsu.edu/tea/

View our code!

https://github.com/jangoflyte/Senior-Project

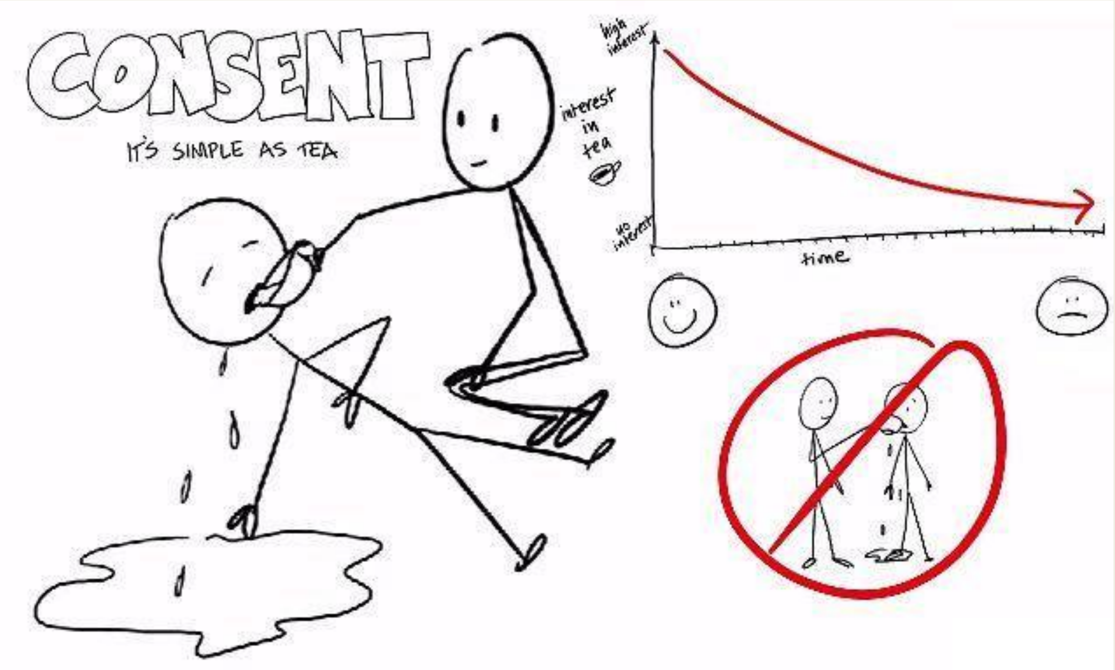
Risk Assessment

- The Requestor must know the requestees contact information
  - Otherwise the application would not function
- The Requestor and requestee must have @broncos.uncfsu.edu emails
  - Otherwise the mail server would not be able to deliver the email



Struggles

- Change of project from original topic of sexual consent because it was a “sensitive topic “ and it required special paperwork.
  - However, we just revised our project to be a scheduler for tea a metaphor for our original topic.



What is Ahead

- Add SMS Messaging
- Added Security for the web application such as email encryption
- Allow web server to mail to other hosts
- Switch to a Google Calendar API or iCloud Calendar API
- Modify or improve logic within the application

Summary

- Introduction
- Background
- Tool & Techniques
- Problem Solution
- Risk Assessment
- Struggles
- What is next