

Personal Portfolio – Khar Jun Shum

Sprint 1 Artefacts

- Stories refinement and prioritisation
 - Evidence

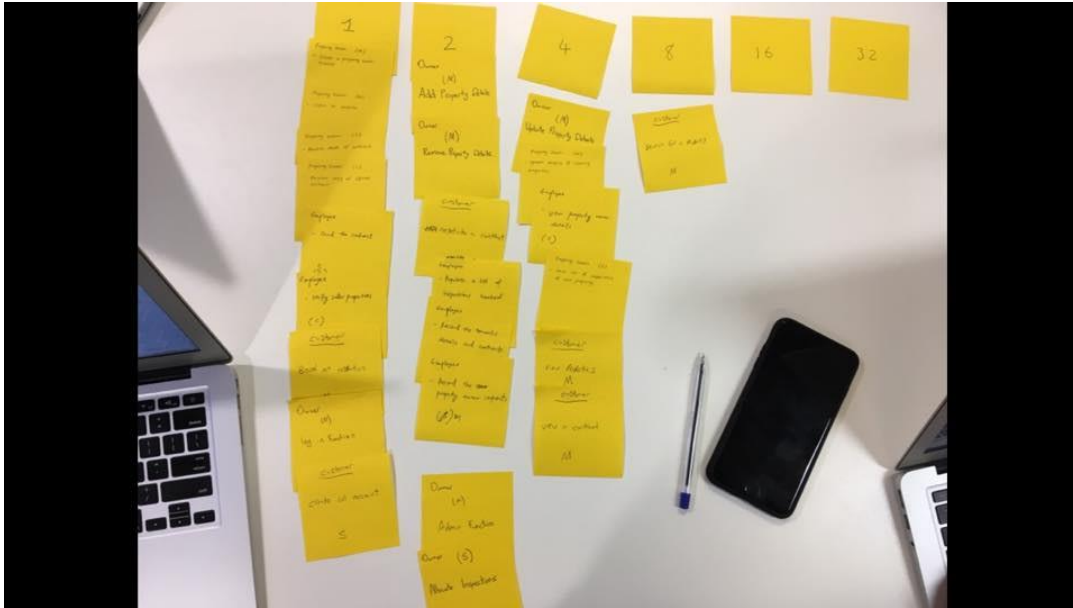


Figure 1: Planning poker for user stories

- Impact of this activity
 - Able to set story points for the stories with the team
 - Good practice as the team needs to come out with an agreement on deciding story points
- Construct 1st sprint plan
 - Evidence

Sprint 1
Total Story Points: 8

Total Hours: 34

Current Velocity: 8

Story 18: View Property Details

Task ID	Task Description	Estimate	Taken
T01	Construct database	2	2
T02	Populate dummy data into database	2	2
T03	Write test cases	2	1
T04	Retrieve dummy data	1	1
T05	Verify story	1	1
Story Points: 2		Total Hours:	7

Figure 2: One of the story in Sprint 1

- Impact of this activity
 - Able to plan a list of tasks which needed to complete the stories
 - Able to estimate the time taken to complete the tasks
 - Record down the real time taken in order to recalculate team velocity in the future
 - Allow team to follow a guideline to implement the website
 - Allow team to check their progress
- Write Codes for user interfaces and backend
 - Evidence

```

properties_controller_test.rb x users_controller_test.rb x users_signup_test.rb x new.html.erb x
header.navbar.navbar-fixed-top.navbar-inverse div.container
1 <% provide(:title, "Log in") %>
2 <header class="navbar navbar-fixed-top navbar-inverse">
3   <div class="container">
4     <%= link_to root_path, id: 'logo' do %>
5       <i class="fa fa-home"></i> David's Rental
6     <% end %>
7   <nav>
8     <ul class="nav navbar-nav navbar-right">
9       <li><%= link_to 'Home', home_path %></li>
10      <li><%= link_to 'Help', help_path %></li>
11      <li><%= link_to 'Log in', login_path %></li>
12    </ul>
13  </nav>
14 </div>
15 </header>
16 <body>
17   <br>
18   <div class="row">
19     <div class="col-md-6 col-md-offset-3">
20       <%= form_for(:session, url: login_path) do |f| %>
21         <%= f.label :email %>
22

```

Figure 3: Codes for designing the user interfaces of login page

```

properties_controller_test.rb x users_controller_test.rb x users_signup_test.rb x new.html.erb x sessions_controller.r
1 class SessionsController < ApplicationController
2   def new
3   end
4
5   def create
6     user = User.find_by(email: params[:session][:email].downcase)
7     if user && user.password == (params[:session][:password])
8       log_in user
9       redirect_to root_url
10    else
11      flash[:danger] = 'Invalid email/password combination!'
12      render 'new'
13    end
14  end
15
16  def destroy
17    log_out
18    redirect_to root_url
19  end
20 end
21

```

Figure 4: Backend code for creating a session when a user login

○ Impact

- Able to complete the implementation of user stories
- Allow the website to fully functional

- Design Database schema and implement
 - Evidence

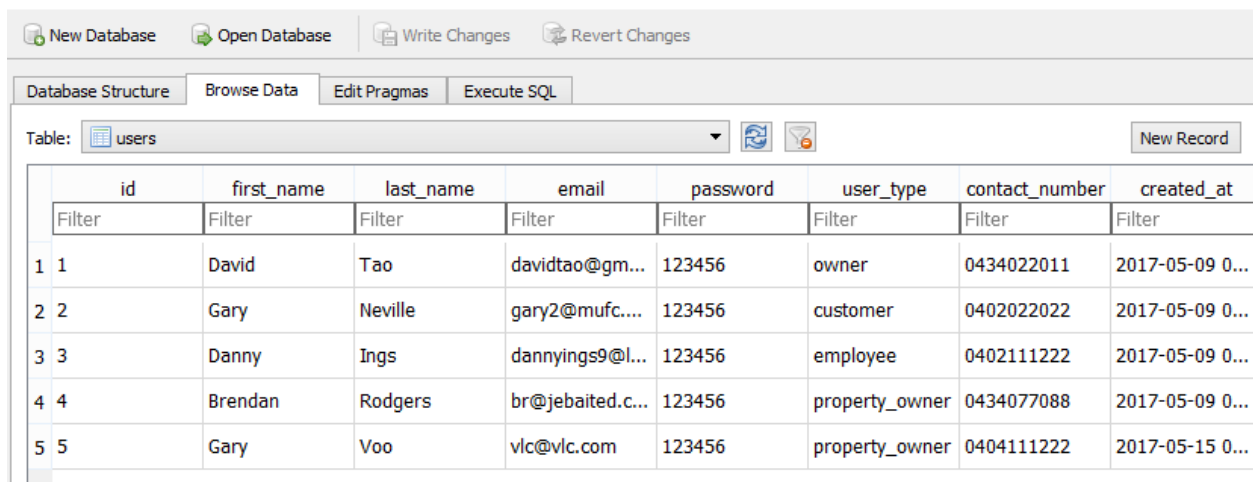


```

1  class CreateUsers < ActiveRecord::Migration[5.0]
2    def change
3      create_table :users do |t|
4        t.string :first_name
5        t.string :last_name
6        t.string :email
7        t.string :password
8        t.string :user_type
9        t.string :contact_number
10
11       t.timestamps
12     end
13   end
14 end

```

Figure 5: Database Schema and migration file to create database



	id	first_name	last_name	email	password	user_type	contact_number	created_at
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	David	Tao	davidtao@gm...	123456	owner	0434022011	2017-05-09 0...
2	2	Gary	Neville	gary2@mufc....	123456	customer	0402022022	2017-05-09 0...
3	3	Danny	Ings	dannying9@l...	123456	employee	0402111222	2017-05-09 0...
4	4	Brendan	Rodgers	br@jebaited.c...	123456	property_owner	0434077088	2017-05-09 0...
5	5	Gary	Voo	vlc@vlc.com	123456	property_owner	0404111222	2017-05-15 0...

Figure 6: The implemented database with users table

- Impact
 - Able to fetch data from the database
 - Able to save data to the database
 - Allow the data to be recorded for multiple functionalities

Sprint 2 Artefacts

- Write Codes for Backend and User Interfaces Design
 - Evidence

```
</body>
<table>
  <tbody>
    <div class="container">
      <div class="row">
        <div class="col-md-10 col-md-offset-1 text-center input-group">
          <%= form_tag(properties_path, :method => 'get') do %>
            <%= text_field_tag :search, params[:search], class: 'form-control', value: '',
              placeholder: "Search Properties (Eg: Tait Street or Kelvin Grove or 4059 or QLD)" %>
            <span class="input-group-btn">
              <%= button_tag(type: 'submit', class: 'btn btn-primary', :name => nil) do %>
                Search <i class="fa fa-search"></i>
              <% end %>
            </span>
          <% end %>
        </div>
      </div>
    </div>
    <br>
    <% if @properties.blank? %>
      <div class="alert alert-danger text-center">
        <h4>There is no property containing the term <%= params[:search] %>.</h4>
      </div>
    </div>
```

Figure 7: Codes for designing a search bar of sprint 2

```
# GET /properties
# GET /properties.json
def index
  @properties = Property.all
  if params[:search]
    @properties = Property.search(params[:search]).order("created_at DESC")
  else
    @properties = Property.all.order("created_at DESC")
  end
end
```

Figure 8: Backend code for searching a property

```
class Property < ApplicationRecord
  def self.search(search)
    where("street ILIKE ? OR suburbs ILIKE ? OR postcode ILIKE ? OR region ILIKE ?", "%#{search}%",
  end
```

Figure 9: Model code for searching a property

- Impact

- Able to complete the implementation of user stories
- Allow the website to fully functional
- Construct Sprint Plan 2 and Plan Tasks
 - Evidence

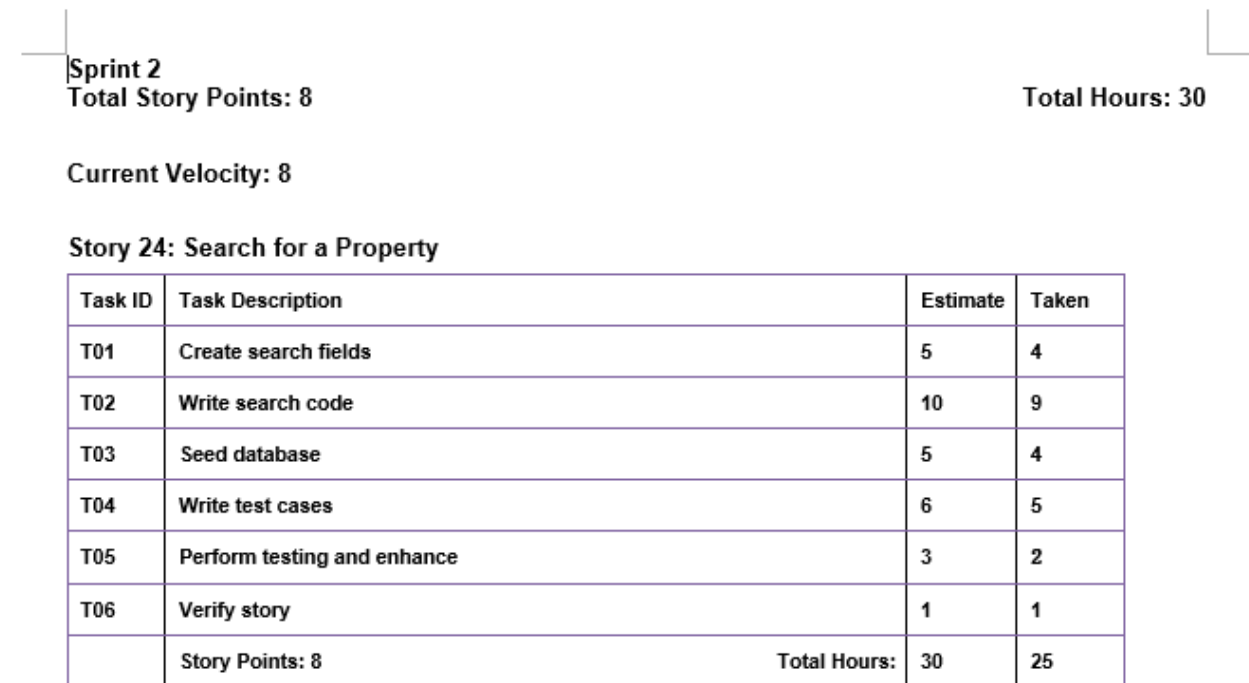


Figure 10: Second Sprint plan

- Impact
 - Able to plan a list of tasks which needed to complete the stories
 - Able to estimate the time taken to complete the tasks
 - Record down the real time taken in order to recalculate team velocity in the future
 - Allow team to follow a guideline to implement the website
 - Allow team to check their progress
- Do Unit Testing
 - Evidence

```

require 'test_helper'

class UsersSignupTest < ActionDispatch::IntegrationTest
  test "invalid signup information for incorrect email format" do
    get new_users_path
    assert_no_difference 'User.count' do
      post users_path, params: { user: { first_name: "Sean",
                                         last_name: "Shum",
                                         password: "123abc",
                                         email: "user@invalid",
                                         contact_number: "0404123123"
                                       } }
    end
    assert_template 'users/new'
  end
end

```

Figure 11: One of the test, testing user signup function

- Impact
 - Able to identify any errors
 - Able to test the website with different scenarios
 - Allow developers to locate potential bugs
- Deploy Website
 - Evidence

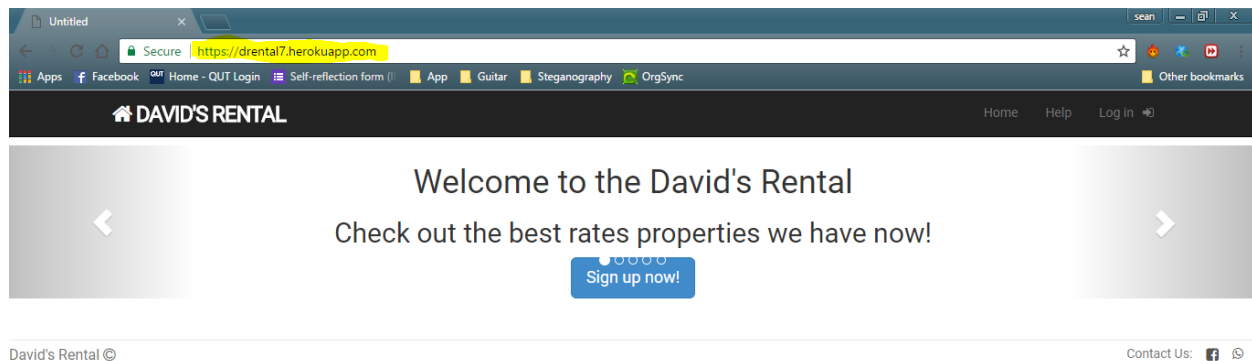


Figure 12: The website deployed to heroku, with the url being highlighted

- Impact
 - Allow users to serve the website in the internet
 - Allow the project team to test the website online