PartyWhip – SENG Project specifications

PartyWhip is an on-line website that allows people to outsource the job of ' *whipping up'* a dinner party, BBQ, birthday party, wedding party, dinner-for-two, a family/friends get-together, just about any occasion where food will be consumed. The website allows people to post requests for some form of catering service, and in response, allows businesses, organisations and private individuals (such as home cooks) to bid on providing the service. PartyWhip is centered around food, and organising that part of an occasion or event that concerns food.

PartyWhip is similar to Airtasker , but the poster's tasks must require the provision of food. Requests for non-food services such as accommodation, flowers and invitations will not be handled.You can look at Airtasker for ideas how 'posters' and 'taskers' do business. PartyWhip is maybe even more closely aligned to hipages , which allows home renovaters to post jobs and trades-people to make offers. Essentially, when a renovater posts a job, 3 trades-people can bid on the job. It is then up to the renovater to accept a bid, possibly after an email exchange with the bidding trades-people.

A bidder on PartyWhip could be the local Domino's who has decided to take home-delivery to the next level and physically cater for special occasions in the local area. It may also be a local aspiring chef who can produce Michelin-star quality food, and is offering to cook in the poster's kitchen or outside BBQ. A family that is requiring food to be cooked and served at home for some special occasion is the target market, but some other location may be offered by the bidder or arranged by the poster. Parameters that the poster may need to stipulate could/should include: date and nature of the occasion/event, the suburb, home or at a location, the level of quality of the food, diet (e.g. vegetarian, vegan, or meat-lover), religious restrictions, the provision of alcohol (maybe it's a wine tasting), the number of people, age profile, big kitchen or small, inside or outside (a BBQ maybe), is there sufficient crockery, glasses, cutlery, even chairs. There are also software parameters such as the maximum number of bids, the closing date for bids, and whether the job has been completed or not.

The advertisement below is an example of a business that would a frequent bidder on your system, but of course businesses will not be allowed to advertise on the system.

The first task that you have is to define a set of requirements. There are some important project considerations that concern achievability and suitability:

- You should be very conscious that you don't have too many requirements: you have only about 8 weeks to build and test the system.
  - To give yourself the best chance of success, use MOSCOW (i.e. prioritise your requirements)
  - You might even consider developing a 'rapid prototype' in (say) 5 weeks and build that into your requirements.
- You need to trade-off 'core-business functionality' and 'creativity'.
  - Core functionality is the natural behaviour that you expect, and defines the minimum, basic behaviour.
  - Creativity is what makes your software innovative, unique. This behaviour should extend the basic system, not replace it.
  - Don't let creativity overwhelm core functionality: identify and separate the core and creative requirements, and again, use MOSCOW
- Scope the system well. I recommend that you do not include detailed requirements for sub-systems such as:
  - Electronic communication between the poster and bidders. This can be very complex to describe and could absorb all your time. Integration with twitter, Facebook or some other service is certainly possible, but it is likely to be out-of-scope for this project, so be careful that you identify what your core business is.
  - Payments. This can also be complex, and is not part of core business. PartyWhip is not a system that concerns itself with money transfers from the poster to the bidder.
  - Accounts. Typically, posters, bidders and others would need to have an account on the system, and these accounts would determine their access privileges. So you will probably need requirements that allow users to log on or off, but be careful that this functionality is kept simple. It is important from a security point of view, but it can also consume much of your time, and it is probably the least interesting aspect of any system.
- The amount of information that the person who posts a request may need to provide could be large. Cut it down to a set of settable parameters that show-off the broad capabilities of your system. Don't make it unnecessarily complicated.
- You cannot do too much verification in the project. Plan for testing and use Dafny wherever you can. Verification will impact the assessment as much as any other aspect of the project (remember this is a workshop in verification). Verifying any sorting and searching algorithms in your system is a minimum. You may also be able to use Dafny during the design process.

Final point: you can call your system something else if you wish.