API contract

We are using Apiary.com as the contract between frontend and backend developers teams.

Testing

Apiary.com makes testing easy for backend. It generates POST, PUT, DELETE request code for bunch of the programming languages: bash (curl), python, java, ruby and so on.
GET requests it is easier to test in the browser.
We test frontend in the browser manually, using EiffelStudio and running EiffelStudio web server locally. And after deployment we check that all is working properly on the cloud hosting.

Project backuping

Code is in google code SVN, we suppose it is secured to hardware problem and has history for software problems.
Deployment infrastructure is in cloud hostings, we suppose cloud hostings is secured for hardware problem.
The only thing that can be changed is Database, and we backup it daily using crontab as scheduler and simple bash script for it.

Email sending

To send email for password recovery, we wrote simple python script that do all the job. Now we need only call it from the eiffel. We did not finish this part yet.

Trello

We use trello for collaborating: https://trello.com/b/uG3PSf4J/dose. Moreover, it is good example of the system similar to the on we are making.

Continuous Integration (CI)

Backend and frontend are integrated together in the cloud hosting.

We have 2 different cloud hostings that are free for students.
One is provided by github.educational and Digital Cloud.
The second is provided by Amazon AWS.
For CI we use selfmade bash scripts, that are supposed to be run manually, after commits or on deadline date.
Scripts are located in the repository:
https://dose2014.googlecode.com/svn/trunk/implementations/group5/deploy/

Production Server

This is cloud server from digital cloud and github.
This server is supposed to have last stable version deployed. Now it is prototype version of 25, November 2014.
Ip address is http://128.199.58.166/.
The script that update production server is "deploy.sh"
It can be run with flags "-f" or "-b" or without any flags. With flag "-f" it deploys only frontend. With flag "-b" it deploys only backend. Without flags, it deploys both.
Deployment is from scratch, all previous files are removed. Only database is not not updated. If we change the scheme of the DB, we should replace it manually.

Development server (CI)

This is cloud server from AWS.
On this server we have latest unstable version deployed. We are trying to update it daily or more often (after big commits).
Ip address is http://54.187.136.201/.
The deployment is the same as for production server, but the script is "dev-deploy.sh".
The only difference, that this server CPU is slower, and deployment from scratch is too long, so here we do incremental deployment, we only do "svn update" and recompile backend and frontend, without removing all the files.

Deployment of the backend

Deployment of the eiffel backend was made considering provided documentation:
https://github.com/kediamanav/Mantra/tree/master/Apache%20configuration%20Files.
So we are using apache as web server, we compile eiffel code with "ec" and put it in "cgi-bin" directory of apache.
All is atomized with bash scripts, that are supposed to be run manually.

Deployment of the frontend

Frontend is deployed using bash scripts too.
To remove vendor libraries from our repository, we are using bower. Bower is nodejs module that download all vendor libraries for you. Also, to check validity of javascript code we use nodejs module "jshint". And to check validity of html markup and to speedup static html requests, we use html+css+js minification (obfuscators). Also to speed up css developing, we are using css preprocessor - less. For working with all this staff, we are using grunt - it is

nodejs module that can run different tasks (check code with jshint, obfuscate html+css+js, compile less).