CODING CONVENTION

The a17s programming standards for enterprise applications will be based on the following practises and convention.

Practices

+ All projects will be started after a group or team review

+ All software development project will have the following documents which will serve their respective purposes as listed:

ITEM	DESCRIPTION
README	*A readme file explaining the purpose and general use of the application. Entries in this document will include installation, contributors, requirement, target system to name a few*
development.txt	*A plain text file holding possible features to be added to future releases of the project. Developers are encouraged to add as many possible features into this file as their thoughts or ideas develop. These features, however, should not be added to the application requirement except approved by the team head or project manager*
lessons.txt	*A plain text file which holds thoughts, considerations, learning points and discoveries of the development team. This file forms the knowledge base of the group*
[app].features	*Set of behave test feature files describing the application expected behaviour and behave tests*
unittest	*Directory of unittest cases for the project*

+ Naming convention is very important. As much as possible realistic names should be given to attributes, objects, classes, databases, tables, models, controllers to name a few. Names assigned to entities should closely follow the application domain. Example, it is preferred to call a medical application user Patient instead of SickPerson or just simply User.

NAMING CONVENTION: Python based applications

ITEM	CONVENTION	EXAMPLE
Database tables	Lowercase separated with underscore, snake, pluralise	user_registrations, waybills, bill_of_landing
Classes	Carmel case	*UserRegistration*
Variables	Lowercase, snake	*user_registration*
Class attributes (methods, property, fields)	Lowercase, snake	*age, gender, date_of_birth*
Functions	Lowercase, snake	*regiseter_user(), do_task()*

+ To ease development, saving time, a lot of effort is put into developing a robust database that is modular and can be extended or reduced to fit the business application needs. Also, effort was made to make these database as generic as possible while also following defined XML schemas that may have been adopted. Developers are to use the core entreprise database designed and deployed for the various business sectors in view. Example, an agro_enterprise application runs on the agricultural_application_db. The same database structure will be used to power the farming grant application, farm monitoring application. The individual application simply makes reference to the tables required ignoring features of the database they will not use.
+ Similar to the approach adopted for managing databases, the front-end of all web and mobile application will be based on the same theme. A default theme will be adopted for all application to ease development and familiarity with web components enabling designers to quickly deploy User Interface.