Xamarin Developer Proficiency Exercise

Objective:
The purpose of this exercise is to assess candidate developer's Xamarin coding knowledge and style. The exercise will be evaluated on coding style, understanding of programming concepts, choice of techniques, and also by the developer's process, as indicated by the trail of git commits.

Specification:
Create a Xamarin based application exhibiting MVVM/MVC pattern which runs on Android devices.
Users should be able to spawn an indefinite number of new views (squares or circles) filled with random colours or pattern images.

No tasks must be changed by the candidate. In case of uncertainty, please contact the employer.

Given time: 8 hours max.

Primary tasks:

- Single view application
- New element is created when user touches the main view background and placed at the touched spot
- The decision between square or circle view is random (code based)
- Created views have random size within adequate range (code based)
- Squares use random images generated by http://www.colourlovers.com/api/patterns/random (imageUrl field)
- Circles use random images generated by http://www.colourlovers.com/api/colors/random (rgb or hex field)
- If no connection available, both squares and circles use code-based random colour generation
- Created views can be dragged with a finger
- Double-tapping a view updates colour/pattern according to the above mentioned rules
- No third-party code/libraries must be used

Challenge tasks (optional):

- Support both phone and tablet form-factor
- View spawning/update animation
- Remove all views by shaking a device
- Custom fancy or animated badge in the action/caption bar displaying a title of the recently used colour/pattern (use "title" field from the XML or hex colour value if using the internal random colour generator)

Evaluation criteria:

- All primary tasks completed.
- Objective oriented design with clearly separated and loosely linked entities (use MVVM, MVC or other classic patterns).
- Good use of caching (file, memory caching; data preloading)
- Usage of data structures
- Use Multithreading/Asynchronicity when needed
- All code should be polished, readable and well commented.

Deliveries:

- Source code and this document with release notes available through GitHub or other public repository. A clear git history with meaningful commit messages is required.
- The source code must compile with no errors.
- The repository must contain a prebuilt Android binary (APK) compiled in Release configuration.
- The application must target Android 4.0 and above.