1 Introduction

1.1 Purpose of application

The application should provide a simple tool to keep track on a student's economy.

1.2 General characteristics of application

The application is providing a default budget and the ability to customize the budget to suit each student personal economy. The application also holds functions for an account with an account balance and the ability to add transactions that has impact on the balance. Both budget and transactions are sorted by categories

2 Test enviroment

The code and .apk for the application can be found at :
https://github.com/daubigne/Android-Budget-Project.git

How to install the application, see the Developer manual
The tester must have or install Eclipse and Android SDK

In addition, the tester must start a test with a new database

2.1 Hardware enviroment

An android mobile device.

2.2 Software environment

Android v2.2, API level 8

3 System information

3.1 System version

Cashin v.0.3

4 Known bugs and limitations

Removing categories:

Removing categories hides all budget-items and transactions from the lists and opening graphs makes the app crash.

Duplicate budget-items when changing category:

If the user edits a budget items category to a category that another budget item already has, those two budget items does not transform into one single budget item.

Budget value disappearing:

When you are in edit mode in the budget tab and write a value to a budget item and then click the ‘add new budget item'-button the written value disappear.

Adding future transaction makes graph work incorrectly:

When adding transactions for the future, the graph stops working correctly.

5 Test specifcation

A document is used to describe our test cases:
Acceptance Test

6 Automatic test

6.1 Code coverage

We use EMMA for checking code coverage. When using EMMA you can’t see how many lines of test code we’ve written.
We aim to cover all the code either by JUnit tests or acceptance test.

6.2 Nightly builds

We do not do nightly builds.

6.3 Unit test

We use JUnit for our unit tests.
We are unable to test activities and fragments through JUnit because they need user input to be tested correctly. We use acceptance testing for theses classes instead.

7 Test report (kan vara bilaga/bilagor)

Test report for version 0.3

Test	**Result**	
Acceptance Test #1	Works	
Acceptance Test #2	Works	
Acceptance Test #3	Works	
Acceptance Test #5	Works	
Acceptance Test #6	Works	
Acceptance Test #7	Works	
Acceptance Test #8	Works	
Acceptance Test #9	Works	
Acceptance Test #10	Doesn’t work.	The application just quits the add transaction-procedure
Acceptance Test #11	Works	