Report

[PUMP:SVAL:100]

This test validates bolus features… blah, blah, blah

* PUMP:SVATR:100

Test 100

[PUMP:SVAL:200]

This test validates basal features… blah, blah, blah

* PUMP:SVATR:200

Test 200

[PUMP:SVAL:300]

This test validates the GUI, including human languages… blah, blah, blah

* PUMP:SVATR:300

Test 300

[PUMP:SVAL:400]

This test validates remaining volume estimation… blah, blah, blah

* PUMP:SVATR:400

Test 400

[PUMP:SVAL:500]

This test validates remaining volume estimation… blah, blah, blah

* PUMP:SVATR:500

Test 500

[PUMP:UNIT:100]

Requirement text not found

* PUMP:UT:100

Unit Test for pump\_aaa.c,h

* PUMP:INS:100

Code Inspection Test for pump\_aaa.c,h

[PUMP:UNIT:110]

Requirement text not found

* PUMP:UT:110

Unit Test for pump\_bbb.c,h

* PUMP:INS:110

Code Inspection Test for pump\_bbb.c,h

[PUMP:UNIT:120]

Requirement text not found

* PUMP:UT:120

Unit Test for pump\_ccc.c,h

* PUMP:INS:120

Code Inspection Test for pump\_ccc.c,h

[PUMP:UNIT:130]

Requirement text not found

* PUMP:UT:130

Unit Test for pump\_ddd.c,h

* PUMP:INS:130

Code Inspection Test for pump\_ddd.c,h

[PUMP:UNIT:140]

Requirement text not found

* PUMP:UT:140

Unit Test for pump\_eee.c,h

* PUMP:INS:140

Code Inspection Test for pump\_eee.c,h

[PUMP:UNIT:150]

Requirement text not found

* PUMP:UT:150

Unit Test for pump\_fff.c,h

* PUMP:INS:150

Code Inspection Test for pump\_fff.c,h

[PUMP:UNIT:160]

Requirement text not found

* PUMP:UT:160

Unit Test for pump\_ggg.c,h

* PUMP:INS:160

Code Inspection Test for pump\_ggg.c,h

[PUMP:UNIT:170]

Requirement text not found

* PUMP:UT:170

Unit Test for pump\_hhh.c,h

* PUMP:INS:170

Code Inspection Test for pump\_hhh.c,h

[PUMP:UNIT:180]

Requirement text not found

* PUMP:UT:180

Unit Test for pump\_iii.c,h

* PUMP:INS:180

Code Inspection Test for pump\_iii.c,h

[PUMP:UNIT:190]

Requirement text not found

* PUMP:UT:190

Unit Test for pump\_jjj.c,h

* PUMP:INS:190

Code Inspection Test for pump\_jjj.c,h

[PUMP:UNIT:200]

Requirement text not found

* PUMP:UT:200

Unit Test for pump\_kkk.c,h

* PUMP:INS:200

Code Inspection Test for pump\_kkk.c,h

[PUMP:UNIT:210]

Requirement text not found

* PUMP:UT:210

Unit Test for pump\_lll.c,h

* PUMP:INS:210

Code Inspection Test for pump\_lll.c,h

[PUMP:UNIT:220]

Requirement text not found

* PUMP:UT:220

Unit Test for pump\_mmm.c,h

* PUMP:INS:220

Code Inspection Test for pump\_mmm.c,h

[PUMP:HRS:100]

The pump shall include a rechargeable Lithium Polymer Battery.

* PUMP:HRD:100

Details regarding the rechargeable Lithium Polymer Battery.

* PUMP:HTP:100

Test 100

[PUMP:HRS:103]

Requirement text not found

* PUMP:HRD:105

Details regarding the fuel gauge hardware for the lithium polymer battery. The battery charge shall be displayed to the user.

[PUMP:HRS:1000]

The pump shall include pressure sensors for use in conjunction with the ideal gas law. The gas law shall be used to estimate remaining insulin volume.

* PUMP:HRD:1000

Details regarding the pressure sensors for use in conjunction with the ideal gas law.

* PUMP:HTP:300

HTP:300 Test 300

[PUMP:HRS:3330]

The pump shall weight no more than 8 ounces dry.

* PUMP:HRD:3330

Details regarding the size and weight of the pump.

* PUMP:HTP:400

HTP:400 Test 400

[PUMP:TBV:1111]

Requirement text not found

* PUMP:HRD:3350

Details regarding the full color touchscreen.

* PUMP:HRD:0000

Details regarding the size and weight of the pump.

[PUMP:PRS:100]

The pump shall include a rechargeable Lithium Polymer Battery.

* PUMP:HRS:100

The pump shall include a rechargeable Lithium Polymer Battery.

[PUMP:PRS:103]

Requirement text not found

* PUMP:HRS:105

The pump shall include fuel gauge hardware for the lithium polymer battery. The battery charge shall be displayed to the user.

[PUMP:PRS:1000]

The pump shall include pressure sensors for use in conjunction with the ideal gas law. The gas law shall be used to estimate remaining insulin volume.

* PUMP:HRS:1000

The pump shall include pressure sensors for use in conjunction with the ideal gas law. The gas law shall be used to estimate remaining insulin volume.

[PUMP:PRS:3330]

The pump shall weight no more than 8 ounces dry.

* PUMP:HRS:3330

The pump shall weight no more than 8 ounces dry.

* PUMP:HRS:3340

The pump shall fit within a volume of 3” by 2” by 0.75”.

[PUMP:PRS:3350]

The pump shall include a full color touchscreen.

* PUMP:HRS:3350

The pump shall include a full color touchscreen.

[PUMP:HRS:105]

The pump shall include fuel gauge hardware for the lithium polymer battery. The battery charge shall be displayed to the user.

* PUMP:HTP:200

HTP:200 Test 200

[PUMP:HTP:300]

HTP:300 Test 300

* PUMP:HTR:300

HTR:300 Test 300

[PUMP:HTP:400]

HTP:400 Test 400

* PUMP:HTR:400

HTR:400 Test 400

[PUMP:HTP:500]

HTP:500 Test 500

* PUMP:HTR:500

HTR:500 Test 500

[PUMP:HTP:1100]

Test 1100

* PUMP:HTR:1100

HTR:1100 Test 1100

[PUMP:HTP:1200]

Test 1200

* PUMP:HTR:1200

HTR:1200 Test 1200

[PUMP:HTP:1300]

Test 1300

* PUMP:HTR:1300

HTR:1300 Test 1300

[PUMP:HTP:1400]

Test 1400

* PUMP:HTR:1400

HTR:1400 Test 1400

[PUMP:HTP:1500]

Test 1500

* PUMP:HTR:1500

HTR:1500 Test 1500

[PUMP:URS:1]

The pump shall provide both a bolus and basal feature.

* PUMP:PRS:1

The pump shall provide both a bolus and basal feature.

[PUMP:RISK:10]

The pump shall deliver no bolus larger than 25 units.

* PUMP:PRS:2

The pump shall limit boluses to not exceed 25 units.

[PUMP:RISK:20]

The pump shall deliver no basal rate larger than 15 units/hr.

* PUMP:PRS:3

The pump shall limit boluses to not exceed 15 units/hour.

[PUMP:URS:3]

The software shall provide a bolus calculator which is supported by programmable correction factors and carb ratios.

* PUMP:PRS:4

The software shall provide a programmable correction factor feature.

* PUMP:PRS:5

The software shall provide a programmable carb ratio feature.

[PUMP:URS:8]

The software shall provide a reverse correction feature for the bolus calculator.

* PUMP:PRS:8

The software shall provide a reverse correction feature for the bolus calculator.

[PUMP:RISK:30]

The bolus calculator shall provide reverse correction.

* PUMP:PRS:8

The software shall provide a reverse correction feature for the bolus calculator.

[PUMP:URS:10]

The software shall provide a means for the user to select between at least five different European human languages.

* PUMP:PRS:10

The software shall provide a means for the user to select between at least five different European human languages.

[PUMP:URS:100]

The pump shall include a rechargeable Battery.

* PUMP:PRS:100

The pump shall include a rechargeable Lithium Polymer Battery.

[PUMP:URS:103]

The pump shall display remaining battery charge.

* PUMP:PRS:105

The pump shall include fuel gauge hardware for the lithium polymer battery. The battery charge shall be displayed to the user.

[PUMP:RISK:40]

The pump shall display remaining battery charge.

* PUMP:PRS:105

The pump shall include fuel gauge hardware for the lithium polymer battery. The battery charge shall be displayed to the user.

[PUMP:URS:1000]

The pump shall display remaining insulin volume.

* PUMP:PRS:1000

The pump shall include pressure sensors for use in conjunction with the ideal gas law. The gas law shall be used to estimate remaining insulin volume.

[PUMP:RISK:50]

The pump shall display remaining insulin volume.

* PUMP:PRS:1000

The pump shall include pressure sensors for use in conjunction with the ideal gas law. The gas law shall be used to estimate remaining insulin volume.

[PUMP:URS:3330]

The pump shall be small and lightweight.

* PUMP:PRS:3330

The pump shall weight no more than 8 ounces dry.

* PUMP:PRS:3340

The pump shall fit within a volume of 3” by 2” by 0.75”.

[PUMP:URS:3350]

The pump shall be easy to use.

* PUMP:PRS:3350

The pump shall include a full color touchscreen.

[PUMP:URS:4000]

The pump shall include an automated dosing algorithm.

* PUMP:PRS:4000

The pump shall include an automated dosing algorithm.

[BOLUS:SRS:1]

The software shall provide a bolus calculator feature.

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:100

This test validates bolus features… blah, blah, blah

[BOLUS:SRS:2]

The bolus size shall be estimated as the sum of the insulin needed to cover the current meal and correct down to 110 mg/dL. If the sum is negative, the bolus is aborted.

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:100

This test validates bolus features… blah, blah, blah

[BOLUS:SRS:5]

The amount of insulin needed to cover the meal shall be estimated only when the user indicates a meal was consumed. The volume of insulin shall be estimated as the number of carbs divided by the CarbRatio.

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:100

This test validates bolus features… blah, blah, blah

[BOLUS:SRS:6]

The amount of insulin needed to correct glucose levels shall be estimated by the following formula if and only if the glucose is greater than 110:

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:100

This test validates bolus features… blah, blah, blah

[BOLUS:SRS:8]

The value of CorrectionBolus shall be set to zero when the glucose is between 80 and 110 mg/dL, inclusive.

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:100

This test validates bolus features… blah, blah, blah

[BOLUS:SRS:12]

The amount of insulin needed to correct glucose levels shall be estimated by the following formula if and only if the glucose is less than 80. Please note that in this case, the dose amount is a negative number.

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:400

This test validates remaining volume estimation… blah, blah, blah

[ACE:SRS:1]

The software shall provide a bolus feature which generates boluses in the range of 0.01 to 25 units, which an increment of 0.01 units.

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:100

This test validates bolus features… blah, blah, blah

[ACE:SRS:5]

The software shall provide a programmable correction factor feature. The software shall support correction factors in the range of 1:10 to 1:600 u/mg/dL.

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:100

This test validates bolus features… blah, blah, blah

[ACE:SRS:6]

The software shall provide a programmable carb ratio feature. The software shall support carb ratios in the range of 1:10 to 1:600 u/g.

* PUMP:SDS:10

Here are details of how the bolus calculator works ….

* PUMP:SVAL:100

This test validates bolus features… blah, blah, blah

[ACE:SRS:2]

The software shall provide a programmable basal rate feature, with basal rates in the range of 0.001 to 15 units/hour in increments of 0.001 u/hr.

* PUMP:SDS:20

Here are details of how the basal rate works ….

* PUMP:SVAL:200

This test validates basal features… blah, blah, blah

[AID:SRS:1]

The dosing algorithm shall include a proportional gain term. The algorithm shall allow proportional gains in the range of 0 to 2.0 in increments of 0.001

* PUMP:SDS:30

Here are details of how the PID algorithm works ….

* PUMP:SVAL:500

This test validates remaining volume estimation… blah, blah, blah

[AID:SRS:2]

The dosing algorithm shall include a derivative gain term. The algorithm shall allow derivative gains in the range of 0 to 2.0 in increments of 0.001

* PUMP:SDS:30

Here are details of how the PID algorithm works ….

* PUMP:SVAL:500

This test validates remaining volume estimation… blah, blah, blah

[AID:SRS:10]

The dosing algorithm shall include an integral gain term. The algorithm shall allow integral gains in the range of 0 to 2.0 in increments of 0.001

* PUMP:SDS:30

Here are details of how the PID algorithm works ….

* PUMP:SVAL:500

This test validates remaining volume estimation… blah, blah, blah

[AID:SRS:12]

The dosing algorithm shall include an integrator window to minimize limit-cycling. The window shall be settable at compile time to a range of 0.002 to 0.01 in increments of 0.001

* PUMP:SDS:30

Here are details of how the PID algorithm works ….

* PUMP:SVAL:500

This test validates remaining volume estimation… blah, blah, blah

[AID:SRS:20]

The dosing algorithm shall include anti-windup reset protection.

* PUMP:SDS:30

Here are details of how the PID algorithm works ….

* PUMP:SVAL:500

This test validates remaining volume estimation… blah, blah, blah

[ACE:SRS:110]

Requirement text not found

* PUMP:SDS:40

Here are details of how the GUI works….

* PUMP:SDS:50

Here are details of how the remaining insulin estimation works… include lots of math ….

[ACE:SRS:120]

Requirement text not found

* PUMP:SDS:40

Here are details of how the GUI works….

* PUMP:SVAL:300

This test validates the GUI, including human languages… blah, blah, blah

[ACE:SRS:10]

The software shall provide a means for the user to select between English, German, French, Spanish, and Czech.

* PUMP:SDS:60

Here are details of how the human languages are swapped around

* PUMP:SVAL:300

This test validates the GUI, including human languages… blah, blah, blah

[ACE:SRS:100]

The GUI shall indicate percent battery charge. The charge shall be displayed in increments of 1%.

* PUMP:SDS:70

Here are details of how the battery estimation works

* PUMP:SVAL:300

This test validates the GUI, including human languages… blah, blah, blah

[PUMP:PRS:1]

The pump shall provide both a bolus and basal feature.

* ACE:SRS:1

The software shall provide a bolus feature which generates boluses in the range of 0.01 to 25 units, which an increment of 0.01 units.

* ACE:SRS:2

The software shall provide a programmable basal rate feature, with basal rates in the range of 0.001 to 15 units/hour in increments of 0.001 u/hr.

* BOLUS:SRS:1

The software shall provide a bolus calculator feature.

* BOLUS:SRS:2

The bolus size shall be estimated as the sum of the insulin needed to cover the current meal and correct down to 110 mg/dL. If the sum is negative, the bolus is aborted.

* BOLUS:SRS:5

The amount of insulin needed to cover the meal shall be estimated only when the user indicates a meal was consumed. The volume of insulin shall be estimated as the number of carbs divided by the CarbRatio.

* BOLUS:SRS:6

The amount of insulin needed to correct glucose levels shall be estimated by the following formula if and only if the glucose is greater than 110:

* BOLUS:SRS:8

The value of CorrectionBolus shall be set to zero when the glucose is between 80 and 110 mg/dL, inclusive.

* BOLUS:SRS:12

The amount of insulin needed to correct glucose levels shall be estimated by the following formula if and only if the glucose is less than 80. Please note that in this case, the dose amount is a negative number.

[PUMP:TBV:1]

Requirement text not found

* ACE:SRS:1

The software shall provide a bolus feature which generates boluses in the range of 0.01 to 25 units, which an increment of 0.01 units.

[PUMP:PRS:5]

The software shall provide a programmable carb ratio feature.

* ACE:SRS:5

The software shall provide a programmable correction factor feature. The software shall support correction factors in the range of 1:10 to 1:600 u/mg/dL.

* BOLUS:SRS:5

The amount of insulin needed to cover the meal shall be estimated only when the user indicates a meal was consumed. The volume of insulin shall be estimated as the number of carbs divided by the CarbRatio.

[PUMP:PRS:6]

Requirement text not found

* ACE:SRS:6

The software shall provide a programmable carb ratio feature. The software shall support carb ratios in the range of 1:10 to 1:600 u/g.

[PUMP:PRS:3]

The pump shall limit boluses to not exceed 15 units/hour.

* BOLUS:SRS:6

The amount of insulin needed to correct glucose levels shall be estimated by the following formula if and only if the glucose is greater than 110:

[PUMP:PRS:8]

The software shall provide a reverse correction feature for the bolus calculator.

* BOLUS:SRS:12

The amount of insulin needed to correct glucose levels shall be estimated by the following formula if and only if the glucose is less than 80. Please note that in this case, the dose amount is a negative number.

[PUMP:PRS:4000]

The pump shall include an automated dosing algorithm.

* AID:SRS:1

The dosing algorithm shall include a proportional gain term. The algorithm shall allow proportional gains in the range of 0 to 2.0 in increments of 0.001

* AID:SRS:2

The dosing algorithm shall include a derivative gain term. The algorithm shall allow derivative gains in the range of 0 to 2.0 in increments of 0.001

* AID:SRS:10

The dosing algorithm shall include an integral gain term. The algorithm shall allow integral gains in the range of 0 to 2.0 in increments of 0.001

* AID:SRS:12

The dosing algorithm shall include an integrator window to minimize limit-cycling. The window shall be settable at compile time to a range of 0.002 to 0.01 in increments of 0.001

* AID:SRS:20

The dosing algorithm shall include anti-windup reset protection.

[PUMP:DER:2]

Requirement text not found

* AID:SRS:1

The dosing algorithm shall include a proportional gain term. The algorithm shall allow proportional gains in the range of 0 to 2.0 in increments of 0.001

* AID:SRS:2

The dosing algorithm shall include a derivative gain term. The algorithm shall allow derivative gains in the range of 0 to 2.0 in increments of 0.001

* AID:SRS:10

The dosing algorithm shall include an integral gain term. The algorithm shall allow integral gains in the range of 0 to 2.0 in increments of 0.001

* AID:SRS:12

The dosing algorithm shall include an integrator window to minimize limit-cycling. The window shall be settable at compile time to a range of 0.002 to 0.01 in increments of 0.001

* AID:SRS:20

The dosing algorithm shall include anti-windup reset protection.

[ACE:SRS:1000]

Requirement text not found

* PUMP:SVAL:300

This test validates the GUI, including human languages… blah, blah, blah